



PROYECTO FIN DE CARRERA PLAN 2000

E.T.S.I.S. TELECOMUNICACIÓN

TEMA: IMAGEN Y TV

TÍTULO: Interacción desde dispositivos Android vía Bluetooth, con juguete teledirigido, para su uso por personas con discapacidad.

AUTOR: Víctor Manuel Lera Martínez

TUTOR: José Manuel Díaz López

Vº Bº.

DEPARTAMENTO: DIAC

Miembros del Tribunal Calificador:

PRESIDENTE: Aurelio Berges García

VOCAL: José Manuel Díaz López

VOCAL SECRETARIO: José Luis Rodríguez Vázquez

DIRECTOR:

Fecha de lectura:

Calificación:

El Secretario,

RESUMEN DEL PROYECTO:

El Proyecto Fin de Carrera, con título, "Interacción desde dispositivos Android vía Bluetooth, con juguete teledirigido, para su uso por personas con discapacidad" pretende completar la primera versión de la aplicación sobre dispositivo Android para manejar un juguete teledirigido añadiendo nuevas formas de interactuar con el dispositivo Android. Para este caso, el juguete teledirigido es el mismo: el robot esférico llamado Sphero. Dicho robot posee una interfaz a través de la cual puede recibir instrucciones, y mediante las cuales, se puede poner en movimiento o iluminarse con diferentes colores. Esta ampliación facilitará la interacción del usuario con el dispositivo Android, además de ampliar la funcionalidad hacia la inversa: recibir los movimientos del robot Sphero en el dispositivo Android cuando es manejado con la mano. Además, completando lo anteriormente escrito, se ha desarrollado una aplicación web que registrará todos los datos de uso del juguete. Estos datos estarán asociados a un usuario, debido a que la aplicación Android debe perfilarse para el uso de un usuario concreto. Por último, se ha realizado un estudio de elementos externos que puedan facilitar la interacción con los dispositivos Android a personas que sufran alguna discapacidad.

El Proyecto Fin de Carrera, con título, "Interacción desde dispositivos Android vía Bluetooth, con juguete teledirigido, para su uso por personas con discapacidad" pretende completar la primera versión de la aplicación sobre dispositivo Android para manejar un juguete teledirigido añadiendo nuevas formas de interactuar con el dispositivo Android. Para este caso, el juguete teledirigido es el mismo: el robot esférico llamado Sphero. Dicho robot posee una interfaz a través de la cual puede recibir instrucciones, y mediante las cuales, se puede poner en movimiento o iluminarse con diferentes colores. Esta ampliación facilitará la interacción del usuario con el dispositivo Android, además de ampliar la funcionalidad hacia la inversa: recibir los movimientos del robot Sphero en el dispositivo Android cuando es manejado con la mano. Completando la primera versión, como cumplimiento de este Proyecto Fin de Carrera, se han realizado una serie de mandos nuevos, los cuales abarcan desde el manejo del robot por instrucciones de voz, movimientos del dispositivo Android desde el que se ejecuta, describir una trayectoria dibujada previamente en el dispositivo Android, o, realizar una serie de movimientos corregidos mediante la aplicación gracias a los límites de movimientos para un usuario concreto que pueden introducirse. Además, completando lo anteriormente escrito, se ha desarrollado una aplicación web que registrará todos los datos de uso del juguete, la cual, explota una base de datos que almacena toda interacción con el juguete. Estos datos estarán asociados a un usuario, debido a que la aplicación Android debe perfilarse para el uso de un usuario concreto. El perfilado de usuario se ha completado añadiendo una serie de información que puede ser útil para la persona que analice el comportamiento de una persona con discapacidad que utilice la aplicación. Por último, se ha realizado un estudio de elementos externos que puedan facilitar la interacción con los dispositivos Android a personas que sufran alguna discapacidad.

The Thesis, titled "Interaction from Android devices via Bluetooth, with remote control toy, for use by people with disabilities" project aims to complete the first version of the application on Android device to manage a remote control toy adding new ways of interacting to Android device. For this case, the remote control toy is the same: the spherical robot called Sphero. This robot has an interface through which it can receive instructions, and by means of which can be set in motion or illuminated with different colors. This expansion will facilitate user interaction with the Android device, and extend the functionality to reverse: receiving Sphero robot movements in the Android device when handled by hand. Completing the first version, in fulfillment of this Thesis, there have been a number of new controls, which range from control of robot by voice instructions, movements Android device from which it runs, describe a path drawn previously on your Android device, or perform a series of movements corrected by applying through limits of movement for a particular user can be made. Besides completing the above written, has developed a web application that will record all data on use of the toy, which exploits a database that stores all interaction with the toy. These data will be associated with a user, because the Android application should be outlined for the use of a particular user. The user profile is completed by adding a range of information that can be useful for the person to analyze the behavior of a disabled person to use the application. Finally, a study was made of external elements that can facilitate interaction with Android devices to people who suffer from a disability.

Índice

| | |
|---|----|
| Índice..... | 1 |
| Capítulo 1: Introducción..... | 5 |
| 1.1 Introducción | 6 |
| 1.2 Objetivos | 8 |
| Capítulo 2: Sistema Operativo Android | 10 |
| 2.1 Introducción a Android | 11 |
| 2.1.1 Historia de Android..... | 12 |
| 2.1.2 Evolución de Android | 13 |
| 2.1.3 Android en el mercado | 36 |
| 2.2.1 Arquitectura..... | 39 |
| 2.2.2 Dalvik..... | 43 |
| 2.2.3 Componentes..... | 47 |
| 2.2.4 Ciclo de vida de una aplicación..... | 50 |
| 2.2.5 Ciclo de vida del componente Activity | 51 |
| 2.2.6 Ciclo de vida de un Service..... | 54 |
| Capítulo 3: Sphero..... | 56 |
| 3.1 Introducción a Sphero | 57 |
| 3.2 Características físicas de Sphero | 57 |
| 3.3 Motivos que llevaron a elegir Sphero..... | 60 |
| 3.4 Datos técnicos del Sphero 2.0 | 61 |
| Capítulo 4: De Flurry a Adaptare | 63 |
| 4.1 Flurry Analytics | 64 |
| 4.2 Adaptare | 65 |
| 4.3 Base de datos..... | 66 |
| 4.4 Inserción en la base de datos | 71 |
| 4.5 Aplicación Web..... | 73 |
| Capítulo 5: Discapacidad | 80 |

| | |
|---|-----|
| 5.1 La comunicación | 81 |
| 5.1.1 Definición..... | 81 |
| 5.1.2 Factores que impiden la comunicación humana..... | 82 |
| 5.1.3 Trastornos en la comunicación y el lenguaje | 83 |
| 5.2 Discapacidades | 86 |
| 5.2.1 Discapacidades psíquicas | 86 |
| 5.2.2 Discapacidades motoras | 88 |
| 5.2.3 Discapacidades sensoriales | 90 |
| 5.3 Sistemas aumentativos y alternativos de comunicación (SAAC) | 93 |
| 5.4 Recursos utilizados en CAA | 94 |
| 5.4.1 Sistemas de símbolos | 94 |
| 5.4.2 Productos de apoyo para la comunicación | 95 |
| 5.4.3 Estrategias y productos de apoyo para el acceso..... | 96 |
| 5.5 Juguetes adaptados | 97 |
| 5.6 Accesibilidad orientada a dispositivos móviles..... | 100 |
| 5.6.1 Accesibilidad para personas con discapacidad auditiva | 100 |
| 5.6.2 Accesibilidad para personas con discapacidad visual | 100 |
| 5.6.3 Accesibilidad para personas mayores..... | 101 |
| Capítulo 6: Dispositivos externos de manejo del juguete | 102 |
| 6. 1 Inconvenientes del uso de Tablets PC o Teléfonos inteligentes..... | 103 |
| 6.2 Posibles soluciones..... | 103 |
| 6.2.1 Joystick..... | 103 |
| 6.2.2 Micrófono..... | 109 |
| 6.2.3 Google Glass | 110 |
| 6.2.4 Teclado para ciegos | 112 |
| 6.2.5 Matamarcianos | 113 |
| Capítulo 7. Aplicación para la accesibilidad Sphero..... | 115 |
| 7.1 Diseño | 116 |

| | |
|---|-----|
| 7.1.1 Especificación de casos de uso..... | 116 |
| 7.1.2 Interfaz de usuario | 119 |
| 7.2 Implementación | 120 |
| 7.2.1 Lenguaje de programación y entorno de desarrollo..... | 120 |
| 7.2.2 Realización del Proyecto | 127 |
| Capítulo 8. Anotaciones sobre la primera versión | 200 |
| 8.1 Solución de errores..... | 201 |
| 8.1.1 Acceso sin datos de usuario | 201 |
| 8.1.2 Acceso sin mandos seleccionados..... | 202 |
| 8.1.3 Botón de añadir mandos | 202 |
| CONCLUSIONES | 203 |
| BIBLIOGRAFÍA | 207 |

Capítulo 1: Introducción

En el primer capítulo de este Proyecto Fin de Carrera se muestra un breve resumen del proyecto realizado, las tecnologías utilizadas y las motivaciones que han llevado al mismo. Además, se comentarán los medios utilizados para obtener los objetivos marcados al comienzo del mismo.

1.1 Introducción

Uno de los grandes problemas con los que se ha encontrado el ser humano a lo largo de los años son las discapacidades, ya sean físicas o psíquicas. Durante este periodo de tiempo se han puesto una serie de remedios para mejorar la vida diaria de las personas que sufren alguna de las mismas, habiéndose avanzado hasta límites en los que los inicios eran impensables. Pero la ciencia y la tecnología, correctamente dirigida por educadores y voluntarios, aún tiene mucho por avanzar. Con este proyecto se pretende aportar un grano de arena más en la mejora de la vida de estas personas.

Dentro de este proyecto nos encontramos con una primera versión, *Interacción desde dispositivos Android vía Bluetooth, con juguete teledirigido, para su uso por personas con discapacidad*, en la que, mediante una aplicación desarrollada para ejecutarse bajo sistema operativo Android, en dispositivos móviles, poder controlar los movimientos del juguete llamado Sphero. Como resultado se obtuvieron un mando joystick virtual, una serie de movimientos ya definidos bajo figuras, una pantalla dividida en porciones de colores que imprimían movimiento al Sphero, y, por último, una pantalla que iluminaba el juguete en función del color seleccionado. Como complemento a esta primera parte, se incluía un menú para el Profesor, donde poder indicar los datos del alumno que utiliza el juguete y se completaba el perfil con diferentes características técnicas del manejo del mismo, además de modificar el listado de mandos que se le mostraban al usuario. Todo este tipo de datos quedarían almacenados en el sistema Flurry, para, en caso necesario, realizar estudios acerca de las estadísticas de uso de la aplicación.

Para la realización de esta segunda versión se pretendió evolucionar la aplicación hacia un lado en el que el manejo del juguete fuese aún más fácil, además de poder aportar otras características que puedan facilitar un poco más el manejo de otros aspectos en el futuro, ya sea aplicando esta tecnología a otras ya existentes, o, simplemente, analizando unos datos más interesantes acerca del uso de la aplicación.

Durante la segunda versión se sigue trabajando con sistema operativo Android, en versiones más actuales, aunque la aplicación puede ejecutarse en versiones anteriores. El juguete Sphero sigue siendo el receptor de las instrucciones, y el dispositivo que lo maneja continúa siendo una Tablet PC, debido a que se considera un objeto más amigable para una persona discapacitada que un teléfono inteligente, simplemente por su tamaño. Por último, se dispone de una aplicación web donde poder consultar las estadísticas de uso de la aplicación.

1.2 Objetivos

Los objetivos de este proyecto pueden dividirse en tres tipos: el primero lúdico, donde el usuario pueda divertirse con más posibilidades del manejo del Sphero; el segundo, educativo, donde el usuario pueda aprender jugando con el Sphero, y, por último, un objetivo analítico, de cara a estudiar los comportamientos de las personas que utilicen la aplicación y con ellos poder seguir investigando en sus necesidades. Sin un orden establecido, estos son los objetivos planteados inicialmente:

- Mejorar el control sobre sus movimientos. En la actualidad es muy complicado su manejo por personas con discapacidad, ya sea motora o intelectual, por lo que sería necesario implementar medidas de apoyo a sus movimientos, que corrigieran automáticamente errores en su manejo.
- Limitar movimientos. Esta funcionalidad debe poderse configurar en el modo Profesor, además de complementar la indicada en el punto anterior.
- Añadido a este tema, y por problemas de manejo de una tablet por personas con problemas de control motor de sus manos y brazos, sería interesante realizar un estudio teórico de dispositivos externos de manejo de movimientos.
- Modificar la funcionalidad de las porciones de colores para que el juguete no se mueva, sino que el usuario pueda marcar una serie de colores y el juguete los repita.
- Realización de una interfaz de usuario sonora utilizando cualquier paquete compatible con Android, como por ejemplo “android.speech”. Ya que cualquier mandato realizado por voz, bastaría para interactuar con un Sphero, sin necesidad de tocar y ver la pantalla del dispositivo Tablet PC.
- Realización de un mando para controlar un Sphero, que únicamente utilice el acelerómetro del dispositivo Tablet PC para dirigir en las diferentes direcciones.

- Eliminar todo lo relacionado con el producto de análisis Flurry, debido a que sus estadísticas son muy genéricas y no aportan la información interesante para los estudios del manejo de la aplicación que puedan llevarse a cabo.
- Para sustituir el producto anterior, realizar una aplicación web que recoja una serie de datos concretos del manejo de la aplicación, los almacene y puedan consultarse a través de la misma.
- Implementación de enlace hacia los ajustes de Android cuando el Bluetooth no esté activado, simulando a la aplicación del fabricante del juguete.
- Crear un mando, capaz de, a partir de la figura dibujada en el layout por el usuario que esté utilizando, enviarla como trayectoria al Sphero, para describirla en la superficie en la que se encuentre.
- Crear un layout, que aprovechando las propiedades de Sphero, se convierta en una pizarra, en la que el usuario pueda dibujar moviendo la Sphero con la mano a modo de ratón.

Capítulo 2: Sistema Operativo Android

Durante este capítulo, siguiendo el orden establecido en la primera versión, se realizará un análisis del sistema operativo Android, mostrando las características del mismo, y actualizando tanto versiones como capacidades añadidas desde la realización de la primera versión.

2.1 Introducción a Android

Android es el sistema operativo de Google orientado a dispositivos móviles, ya sean Tablet PC o teléfonos inteligentes, aunque también pueden encontrarse en relojes inteligentes, televisores y automóviles. Basado en una versión modificada del kernel de Linux 2.6, fue desarrollado por la empresa Android Inc. con apoyo por parte de Google, quien adquirió la empresa en el año 2005. Presentado en el año 2007, el primer producto comercial que llevaba instalado este sistema operativo se puso a la venta en octubre de 2008.

Android es una plataforma de código abierto, y como tal, su distribución es libre y posibilita el acceso y modificación de su código fuente. Distribuida bajo licencia Apache 2.0, fue desarrollado por Google de forma inicial, para más tarde unirse a la Open Handset Alliance (de la cual, Google también forma parte) que está integrada por T-Mobile, Intel, Samsung, HTC o Nvidia entre otros. Incluso formando parte de la alianza anteriormente citada, Google ha sido la compañía que ha publicado la mayor parte del código fuente bajo la licencia Apache.



Figura 2.1 – Teléfono inteligente junto a la imagen de Android

Otra de las características importantes del sistema operativo Android es el elevado número de desarrolladores programando aplicaciones, las cuales luego pueden terminar en los dispositivos a través de sus plataformas de distribución (Google Play). Se estima que a mediados del año 2014 el número de aplicaciones escritas para Android ha alcanzado el millón (de las cuales, dos tercios son gratuitas, además de más baratos con respecto a uno de sus principales competidores, Apple). Dentro de estos cálculos, no se tienen en cuenta la serie de aplicaciones que se pueden obtener en tiendas no oficiales, es decir, en otros sitios diferentes a la tienda oficial de aplicaciones Google Play.

Para poder desarrollar software bajo para sistema operativo Android, a los programadores se les ofrece, de forma gratuita, el SDK (software development kit) y un plugin para integrar dentro del entorno de desarrollo Eclipse, en el cual, se incluyen todas las APIs necesarias. Además, se ofrece también, un entorno propio de desarrollo llamado Android Studio, con el cual se puede realizar el desarrollo de aplicaciones de la misma forma que con Eclipse. Ambos entornos poseen un emulador de los diferentes dispositivos Android que se encuentran en el mercado, facilitando las pruebas del software desarrollado.

En el año 2014, Google pretende unificar tanto el software como el hardware, por lo que han sido presentados productos como Android TV, Android Auto, Android Wear o Android One.

2.1.1 Historia de Android

Tal y como se apunta en la introducción, Android es un sistema operativo desarrollado por la empresa Android Inc bajo apoyo financiero por Google, quien además, en el año 2005, adquiere la empresa. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Dicho conglomerado está compuesto por 84 compañías dedicadas a realizar estándares abiertos para dispositivos móviles. Entre las cuales se encuentran los principales fabricantes de teléfonos inteligentes o diferentes sistemas electrónicos.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, no siendo posible utilizarlo de forma comercial hasta octubre 2008 dentro del modelo de teléfono inteligente HTC Dream. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

Durante su corta historia, Android ha sufrido numerosas evoluciones y versiones, siendo las mismas para corrección de errores, o, simplemente, ampliar el catálogo de funciones a disponer por parte del usuario. Este número elevado de actualizaciones, las cuales no siempre pueden utilizarse correctamente en todos los fabricantes, hacen que Google sea considerado uno de los elementos promotores de la obsolescencia programada.

2.1.2 Evolución de Android

La evolución de Android desde su nacimiento hasta nuestros días ha sido constante, publicando versiones con un buen número de características nuevas, llegando a ser alguna de ellas una evolución radical frente a la versión predecesora. Sus nuevas versiones aparecen cada cortos periodos de tiempo, no siempre adaptándose al hardware disponible hasta la fecha.

Android actualmente es el sistema operativo móvil líder, tanto en cuota de mercado, como en innovación, y a falta de conocer la nueva versión Android 5.0, la cual no se ha presentado oficialmente pero sí se ha filtrado algún detalle que podría disponer la misma, se va a exponer mediante un resumen de las versiones desarrolladas, que ayudarán a la comprensión de cómo poco a poco, ha llegado a ser un referente en cuanto a modelo de sistema operativo.

A continuación, se mostrará información relacionada con todas las versiones de Android.

Android Beta

La primera versión de Android en modo Beta se hizo pública el 5 de noviembre de 2007. Mientras que el SDK (software development kit) fue lanzado el 12 de ese mismo mes. Las versiones públicas Beta del SDK fueron las siguientes (junto a la fecha de publicación):

- 16 de noviembre de 2007: m3-rc22a
- 14 de diciembre de 2007: m3-rc37a
- 13 de febrero de 2008: m5-rc14
- 3 de marzo de 2008: m5-rc15
- 18 de agosto de 2008: 0.9
- 23 de septiembre de 2008: 1.0-r1

Android 1.0 Apple Pie

La primera versión comercial de Android fue lanzada el 23 de septiembre de 2008, dentro del dispositivo móvil HTC Dream. Fue llamada Apple Pie (tarta de manzana) y, a partir de entonces, todas sus versiones llevaron un nombre específico más allá de la numeración que indica la versión. Sus principales características fueron las siguientes:

- Android Market Programa con un mercado para la descarga y actualización de aplicaciones.
- Navegador Web para visualizar páginas webs en full HTML y XHTML – múltiples páginas mostradas como ventanas ("tarjetas").
- Soporte Cámara – sin embargo esta versión carece de la opción de cambiar la resolución de la cámara, balance de blancos, calidad, etc.
- Carpetas permiten la agrupación de un número de iconos de aplicaciones dentro de una simple carpeta en la pantalla de inicio.
- Acceso a servidores de correo electrónico por web, soporte POP3, IMAP4 y SMTP.
- Sincronización de Gmail con la aplicación de Gmail.
- Sincronización de Google Contacts aplicación de personas.
- Sincronización de Google Calendar con la aplicación de calendario.

- Google Maps con Latitude y Street View para ver mapas e imágenes por satélite, así como para encontrar negocios locales y obtener direcciones de conducción usando GPS.
- Google Sync, permite la administración de la sincronización OTA de Gmail, Personas, y Calendario.
- Google Search, permite a los usuarios buscar en internet, en aplicaciones del teléfono móvil, en contactos, en calendario, etc.
- Mensajería instantánea Google Talk.
- Mensajería instantánea, mensajes de texto y MMS.
- Reproductor de medios, habilitada administración, importación, y reproducción de archivos multimedia – sin embargo, esta versión carece de soporte de vídeo y estéreo por Bluetooth.
- Las notificaciones aparecen en la barra de estado, con opciones para configurar alertas por timbre, LED o vibración.
- Marcación por voz permite marcar y llamar sin escribir nombre o número.
- Fondo de escritorio permite al usuario configurar una imagen de fondo o una foto detrás de los iconos y widgets de la pantalla de inicio.
- Reproductor de vídeo YouTube.
- Otras aplicaciones incluyen: Alarma, Calculadora, Marcación (teléfono), Pantalla de inicio (launcher), Imágenes (Galería) y ajustes.
- Soporte para Wi-Fi y Bluetooth.

Android 1.1 Banana Bread

La actualización Android 1.1 Banana Bread (Pan de plátano) fue lanzada el 9 de febrero de 2009. Al igual que la versión anterior, fue lanzada para el teléfono inteligente HTC Dream. Dentro de la compañía fue conocido como Petit Four, aunque comercialmente fue lanzado como Banana Bread. La corrección de fallos, actualización de la API y añadir una serie de características fueron las motivaciones de esta actualización. A continuación las mejoras incluidas en esta versión:

- Detalles y reseñas disponibles cuando un usuario busca negocios en los mapas.
- Pantalla en llamada más larga por defecto cuando están en uso el manos libres, además la habilidad de mostrar/esconder el marcador.
- Posibilidad de guardar archivos adjuntos en los mensajes.
- Añadido soporte para marquesina en diseños de sistemas.

Android 1.5 Cupcake

Durante la primera de 2009 (30 de abril) fue publicada esta siguiente versión, conocida como Cupcake. Basada en el núcleo de Linux 2.6.27, incorporó las siguientes correcciones y evolutivos:

- Soporte para teclados virtuales de terceros con predicción de texto y diccionario de usuarios para palabras personalizadas.
- Soporte para Widgets - vistas de miniaturas de las aplicaciones que pueden ser incrustadas en otras aplicaciones (tal como la pantalla inicio) y recibir actualizaciones periódicas.
- Grabación y reproducción en formatos MPEG-4 y 3GP.
- Auto-sincronización y soporte para Bluetooth estéreo añadido (perfiles A2DP y AVRCP).
- Características de Copiar y pegar agregadas al navegador web.
- Fotos de los usuarios son mostradas para favoritos en los contactos.
- Marcas de fecha/hora mostradas para eventos en registro de llamadas y acceso. con un toque a la tarjeta de un contacto desde un evento del registro de llamadas.
- Pantallas de transiciones animadas.
- Agregada opción de auto-rotación.
- Agregada la animación de inicio por defecto actual.
- Habilidad de subir vídeos a YouTube.
- Habilidad de subir fotos a Picasa.

Android 1.6 Donut

A finales del verano de 2009, concretamente el 15 de septiembre, fue publicado el SDK de la versión 1.6 Donut, basado en el núcleo Linux 2.6.29. Incluía las siguientes mejoras:

- Mejora en la búsqueda por entrada de texto y voz para incluir historial de favoritos, contactos y la web.
- Habilidad de los desarrolladores de incluir su contenido en los resultados de búsqueda.
- Motor multi-lenguaje de síntesis de habla para permitir a cualquier aplicación de Android "hablar" una cadena de texto.
- Búsqueda facilitada y habilidad para ver capturas de las aplicaciones en el Android Market (Google Play).
- Galería, cámara y videocámara con mejor integración, con rápido acceso a la cámara.
- La galería ahora permite a los usuarios seleccionar varias fotos para eliminarlas.
- Actualización soporte a tecnología para CDMA/EVDO, 802.1x, VPNs y un motor text-to-speech.
- Soporte para resoluciones de pantalla WVGA.
- Mejoras de velocidad en búsqueda y aplicaciones de cámara.
- Framework de gestos ampliado y una nueva herramienta de desarrollo GestureBuilder.

Android 2.0/2.1 Eclair

El 26 de octubre de 2009 fue publicado el SDK de Android 2.0, basado también en el núcleo de Linux 2.6.29. Las versiones 2.0.1 y 2.1 también fueron conocidas como Eclair. Dichas versiones fueron lanzadas el 3 de diciembre del mismo año y el 12 de enero de 2010 respectivamente. Para estas versiones se incorporaron las siguientes características:

2.0

- Sincronización cuenta expandida, permitiendo a los usuarios agregar múltiples cuentas al dispositivo para sincronización de correo y contactos.
- Soporte intercambio de correo, con bandeja combinada para buscar correo desde múltiples cuentas en la página.

- Soporte Bluetooth 2.1.
- Habilidad para tocar un foto de un contacto y seleccionar llamar, enviar SMS o correo a la persona.
- Habilidad para en todos los mensajes SMS y MMS guardados, con eliminación de mensajes más antiguos en una conversación automáticamente cuando un límite definido se ha alcanzado.
- Nuevas características para la cámara, incluyendo soporte de flash, zoom digital, modo escena, balance de blancos, efecto de colores y enfoque macro.
- Mejorada velocidad de tipeo en el teclado virtual, con diccionario inteligente que aprende el uso de palabras e incluye nombres de contactos como sugerencias.
- Renovada interfaz de usuario del navegador con imágenes en miniatura de marcador, zoom de toque-doble y soporte para HTML5.
- Vista agenda del calendario mejorada, que muestra el estado asistiendo a cada invitado, y la capacidad de invitar a nuevos invitados a los eventos.
- Optimización en velocidad de hardware y GUI renovada.
- Soporte para más tamaños de pantalla y resoluciones, con mejor ratio de contraste.
- Mejorado Google Maps 3.1.2.
- Clase MotionEvent mejorada para rastrear eventos multi-touch.
- Adición de fondos de pantalla animados, permitiendo la animación de imágenes de fondo de la pantalla inicio para mostrar movimiento.

2.0.1

- Cambios menores de API, arreglo de errores y cambios del comportamiento del framework.

2.1

- Modificaciones menores de la API y correcciones de errores.

Android 2.2.x Froyo

El SDK del Android 2.2 Froyo (yogur helado) fue lanzado el 20 de mayo de 2010, basado en el núcleo de Linux 2.6.32. A esta nueva versión le siguieron otras tres con la misma denominación, publicadas el 18 de enero de 2011, 22 de enero de 2011 y 21 de noviembre de 2011. A continuación se detallan las características de cada una de ellas:

2.2

- Optimizaciones en velocidad, memoria y rendimiento.
- Mejoras adicionales de rendimiento de aplicación, implementadas mediante compilación Just-in-time (JIT).
- Integración del motor de JavaScript V8 de Chrome en el navegador.
- Soporte para el servicio Android Cloud to Device Messaging (C2DM), habilitando notificaciones push.
- Soporte para Microsoft Exchange mejorado, incluyendo políticas de seguridad, auto-descubrimiento, consulta a la Global Access List (GAL), sincronización de calendario, y borrado remoto.
- Mejoras en la aplicación del lanzador con accesos directos de las aplicaciones teléfono y navegador web.
- Funcionalidad de anclaje de red por USB y Wi-Fi hotspot.
- Agregada opción para deshabilitar acceso de datos sobre red móvil.
- Actualizada la aplicación Market con características de grupo y actualizaciones automáticas.
- Cambio rápido entre múltiples lenguajes de teclado y diccionario.
- Marcación por voz e intercambio de contactos por Bluetooth.
- Soporte para docks Bluetooth-habilitado para autos y de escritorio.
- Soporte para contraseñas numéricas y alfanuméricas.
- Soporte para subida de archivos en la aplicación del navegador.
- Soporte para instalación de aplicaciones en la memoria expandible.
- Soporte para Adobe Flash.
- Soporte para pantallas de alto número de PPI (320 ppi), como 4" 720p.
- Galería permite a los usuarios ver pilas de imágenes mediante un gesto de zoom.

2.2.1

- Arreglo de errores, actualizaciones de seguridad y mejoras de rendimiento.

2.2.2

- Arreglo de fallos menores, incluyendo problemas con el ruteo de SMS que afectaron al Nexus One.

2.2.3

- Dos parches de seguridad.

Android 2.3.x Gingerbread

El SDK de Android 2.3 Gingerbread (Pan de Jengibre), basado en el núcleo Linux 2.6.35, fue lanzado el 6 de diciembre de 2010. Como en las versiones anteriores, se lanzaron varias actualizaciones bajo el mismo nombre. Las primeras fueron la 2.3.0 y 2.3.1, lanzadas en la fecha indicada anteriormente. Posteriormente salieron la 2.3.2 en enero de 2011, 2.3.3 el 9 de febrero de 2011, 2.3.4 el 28 de abril de 2011, 2.3.5 el 25 de julio de 2011, 2.3.6 el 2 de septiembre de 2011 y, por último, 2.3.7 el 21 de septiembre de 2011. Cada una de las versiones incorporó las siguientes mejoras:

2.3.0/2.3.1

- Actualizado el diseño de la interfaz de usuario con incrementos en velocidad y simpleza.
- Soporte para tamaños y resoluciones de pantalla extra-grandes (WXGA y mayores).
- Soporte nativo para SIP y telefonía por internet VoIP.
- Entrada de texto del teclado virtual más rápida e intuitiva, con mejoras en precisión, texto sugerido y entrada por voz.
- Mejoras en la funcionalidad de copiar/pegar, permitiendo a los usuarios seleccionar una palabra al presionar-mantener, copiar y pegar.

- Soporte para Near Field Communication (NFC), permitiendo al usuario leer la etiqueta NFC incrustada en un póster, sticker o anuncio publicitario.
- Nuevos efectos de audio tales como reverberación, ecualizador, virtualización de audífonos y aumento de bajos.
- Nuevo gestor de descargas, que da a los usuarios fácil acceso a cualquier archivo descargado del navegador, correo electrónico u otra aplicación.
- Soporte para múltiples cámaras en el dispositivo, incluyendo cámara frontal-facial, si está disponible.
- Soporte para reproducción de video por WebM/VP8, codificación de audio por AAC.
- Mejoras en la administración de la energía, con un mayor rol activo en aplicaciones de administración que se mantienen activas en el dispositivo por mucho tiempo.
- Mejorado soporte para el desarrollo de código nativo.
- Cambio desde YAFFS a ext4 en dispositivos nuevos.
- Mejoras en audio, gráficos y entrada para desarrolladores de juegos.
- Recolector basura concurrente para incrementar el rendimiento.
- Soporte nativo para más sensores (tales como giroscopio y barómetro).

2.3.2

- Mejoras y errores solucionados para el modelo Google Nexus S.

2.3.3

- Varias mejoras y arreglos del API.

2.3.4

- Rebaja de la seguridad de SSL al usar protocolos de cifrado inseguros.
- Soporte de chat de video o voz, usando Google Talk.
- Soporte a la biblioteca Open Accessory. Open Accessory fue introducida en 3.1 (Honeycomb) pero la biblioteca Open Accessory Library subvenciona en 2.3.4 agregado su soporte cuando un periférico USB es conectado con software compatible y una aplicación compatible en el dispositivo.

2.3.5

- Mejoras en el sistema.
- Mejoras en el rendimiento por red del Nexus S 4G.
- Arreglado una falla de Bluetooth en el Samsung Galaxy S.
- Mejoras a la aplicación de correo electrónico.
- Animación de sombras al deslizar por listas.
- Mejoras al software de la cámara.
- Mejorada la eficiencia de la batería.

2.3.6

- Arreglado fallo en la búsqueda por voz.

La actualización 2.3.6 tuvo el efecto secundario de menoscabar la funcionalidad de Wi-Fi hotspot de muchos teléfonos Nexus S canadienses. Google reconoció este problema y fue arreglado a finales de septiembre.

2.3.7

- Soporte de Google Wallet para el Nexus S 4G. Esta versión es exclusiva para usuarios en Canadá.

Android 3.x Honeycomb

El SDK de Android 3.0 Honeycomb (Panal de Miel), lanzado exclusivamente para dispositivos Tablet PC, fue lanzado al mercado el 22 de febrero de 2011. Está basada en el núcleo de Linux 2.6.36, y el primer dispositivo que lo incluyó fue el Motorola Xoom, lanzado el 24 de febrero de ese mismo año. Además de la versión 3.0 ya comentada, tiene otras seis versiones posteriores: 3.1, lanzada el 10 de mayo de 2011, 3.2, lanzada el 15 de julio de 2011, 3.2.1, lanzada el 20 de septiembre de 2011, 3.2.2, lanzada el 30 de agosto de 2011, 3.2.3, lanzada el 1 de diciembre de 2011 y 3.2.4 lanzada el 28 de febrero de 2012. Las características, por versión, son las siguientes:

3.0

- Soporte optimizado para tablets, con una nueva y "virtual" interfaz de usuario holográfica.
- Agregada barra de sistema, con características de acceso rápido a notificaciones, estados y botones de navegación suavizados, disponible en la parte inferior de la pantalla.
- Añadida barra de acción (Action Bar en inglés), entregando acceso a opciones contextuales, navegación, widgets u otros tipos de contenido en la parte superior de la pantalla.
- Multitarea simplificada – tocando Aplicaciones recientes en la barra del sistema permite a los usuarios ver instantáneas de las tareas en curso y saltar rápidamente de una aplicación a otra.
- Teclado rediseñado, permitiendo una escritura rápida, eficiente y acertada en pantallas de gran tamaño.
- Interfaz simplificada y más intuitiva para copiar/pegar.
- Las pestañas múltiples reemplazan las ventanas abiertas en el navegador web, además de la característica de auto completado texto y un nuevo modo de "incógnito" permitiendo la navegación de forma anónima.
- Acceso rápido a las características de la cámara como la exposición, foco, flash, zoom, cámara facial-frontal, temporizador u otras.
- Habilidad para ver álbumes y otras colecciones de fotos en modo pantalla completa en galería, con un fácil acceso a vistas previas de las fotografías.
- Nueva interfaz de contactos de dos paneles y desplazamiento rápido para permitir a los usuarios organizar y reconocer contactos fácilmente.
- Nueva interfaz de correo de dos paneles para hacer la visualización y organización de mensajes más eficiente, permitiendo a los usuarios seleccionar uno o más mensajes.
- Soporte para videochat usando Google Talk.
- Aceleración de hardware.
- Soporte para microprocesadores multi-núcleo.
- Habilidad para encriptar todos los datos del usuario.
- Mejoras en el uso de HTTPS con Server Name Indication (SNI).
- Sistema de archivos en el espacio de usuario (FUSE; kernel module).

3.1

- Refinamiento a la interfaz de usuario.
- Conectividad para accesorios USB.
- Lista expandida de aplicaciones recientes.
- Widgets redimensionables en la pantalla de inicio.
- Soporte para teclados externos y dispositivos punteros.
- Soporte para joysticks y gamepads.
- Soporte para reproducción de audio FLAC.
- Bloqueo de Wi-Fi de alto rendimiento, manteniendo conexiones Wi-Fi de alto rendimiento cuando la pantalla del dispositivo está apagada.
- Soporte para proxy HTTP para cada punto de acceso Wi-Fi conectado.

3.2

- Mejoras de soporte de hardware, incluyendo optimizaciones para un amplio rango de tablets.
- Incrementada la capacidad de las aplicaciones para acceder a archivos de las tarjetas SD, por ejemplo para sincronización.
- Modo de vista de compatibilidad para aplicaciones que no han sido optimizadas para resoluciones de pantalla de tablets.
- Nuevas funciones de soporte de pantalla, dando a los desarrolladores un mayor control sobre la apariencia de la pantalla en diferentes dispositivos Android.

3.2.1

- Corrección de errores menores y mejoras de seguridad, mejoras en estabilidad y Wi-Fi.
- Actualización del Android Market con actualizaciones texto de términos y condiciones de fácil lectura.
- Actualización de Google Books.
- Mejoras en el soporte de Adobe Flash del navegador.
- Mejoras en la predicción de escritura a mano en chino.

3.2.2

- Arreglo de fallos y otras mejoras menores para el Motorola Xoom 4G.

3.2.3

- Soporte para "Pay as You Go" para tabletas 3G y 4G.

3.2.4

- Arreglado problema de conectividad de datos en modo avión en la versión estadounidense 4G del Motorola Xoom.

Android 4.0.x Ice Cream Sandwich

Basado en el núcleo de Linux 3.0.1, el SDK para Android 4.0.0 Ice Cream Sandwich (Sandwich de Helado) fue lanzado al mercado el 19 de octubre de 2011. Según Gabe Cohen, Product Manager de Google, Ice Cream Sandwich era teóricamente compatible con cualquier dispositivo Android 2.3 que estuviera en el mercado en ese momento, siempre y cuando el procesador y la memoria RAM del dispositivo fueran las mínimas exigibles por esta nueva versión de Android.

El código fuente de esta versión fue publicado el 14 de noviembre de ese año. Además de la versión 4.0.0, existen otras cuatro versiones: 4.0.1, compartiendo fecha de lanzamiento (apenas dos días después) y características con su predecesora, 4.0.2, lanzada el 29 de noviembre de 2011, 4.0.3, lanzada el 16 de diciembre de 2011 y 4.0.4, publicada el 8 de noviembre de 2012. A continuación, las características de cada una de estas versiones:

4.0.0

- Botones suaves Android 3.x están ahora disponibles para usar en los teléfonos móviles.
- Separación de widgets en una nueva pestaña, listados de forma similar a las aplicaciones.
- Facilidad para crear carpetas, con estilo de arrastrar y soltar.
- Lanzador personalizable.

- Buzón de voz mejorado con la opción de acelerar o retrasar los mensajes del buzón de voz.
- Funcionalidad de pinch-to-zoom en el calendario.
- Captura de pantalla integrada (manteniendo presionado los botones de bloqueo y de bajar volumen).
- Corrector ortográfico del teclado mejorado.
- Habilidad de acceder a aplicaciones directamente desde la pantalla de bloqueo.
- Funcionalidad copiar-pegar mejorada.
- Mejor integración de voz y dictado de texto en tiempo real continuo.
- Desbloqueo facial, característica que permite a los usuarios desbloquear los equipos usando software de reconocimiento facial.
- Nuevo navegador web con pestañas bajo la marca de Google Chrome, permitiendo hasta 15 pestañas.
- Sincronización automática del navegador con los marcadores de Chrome del usuario.
- Nueva tipografía para la interfaz de usuario, Roboto.
- Sección para el uso de datos dentro de la configuración que permite al usuario poner avisos cuando se acerca a cierto límite de uso, y desactivar los datos cuando se ha excedido dicho límite.
- Capacidad para cerrar aplicaciones que están usando datos en segundo plano.
- Aplicación de la cámara mejorada sin retardo en el obturador, ajustes para el time lapse, modo panorámico y la posibilidad de hacer zoom durante la grabación.
- Editor de fotos integrado.
- Nuevo diseño de la galería, organizada por persona y localización.
- Aplicación "People" actualizada con integración en redes sociales, actualización de estados e imágenes en alta resolución.
- Android Beam, una característica de Near Field Communication que permite el rápido intercambio de corto alcance de enlaces web favoritos de un navegador de internet, información de contactos, direcciones, vídeos de YouTube y otros datos.
- Soporte para el formato de imagen WebP.
- Aceleración por hardware de la interfaz de usuario.
- Wi-Fi Direct.

- Grabación de vídeo a 1080P para dispositivos con Android de serie.
- Android VPN Framework (AVF), y el módulo de kernel TUN. Antes de esta versión, el software VPN requería ser root de Android.

4.0.1

- Arreglados fallos menores en el Samsung Galaxy Nexus.

4.0.2

- Arreglados fallos menores en el Galaxy Nexus de Verizon, el lanzamiento en EE.UU. fue retrasado hasta diciembre de 2011.

La publicación de la versión 4.0.2 tenía un bug en el Galaxy Nexus que hacía fallar la aplicación market (Google Play a día de hoy) cuando los usuarios trataban de ver los detalles de cualquier aplicación Android. Además, reducía capacidades del NFC (Near Field Communication) del teléfono Nexus. Estos fallos afectaron principalmente a las versiones lanzadas en Canadá.

4.0.3

- Numerosas optimizaciones y corrección de errores.
- Mejoras en gráficos, bases de datos, corrección ortográfica y funcionalidades Bluetooth.
- Nueva API para los desarrolladores, incluyendo una API de actividad social en el proveedor de contactos.
- Mejoras en el calendario.
- Nuevas aplicaciones de la cámara en mejora de la estabilidad en los videos y resolución QVGA.
- Mejoras de accesibilidad tales como la mejora de acceso al contenido para lectores de pantalla.

4.0.4

- Mejoras de estabilidad.
- Mejor rendimiento de la cámara.
- Rotación de la pantalla más fluida.
- Mejoras en el reconocimiento de los números en el teléfono.

Android 4.1 Jelly Bean

El 27 de junio de 2012 Google fue anunciada la versión 4.1 Jelly Bean (Gominola) de Android. Basado en el núcleo de Linux 3.0.31, el principal objetivo de esta nueva entrega de Android era mejorar la funcionalidad y el rendimiento de la interfaz de usuario, gracias, esto último, al "Proyecto Butter", el cual usa anticipación táctil, triple buffer, latencia vsync extendida y un arreglo en la velocidad de cuadros de 60 fps para crear una fluida y "mantecosa"-suavidad de la interfaz de usuario. Jelly Bean fue lanzado bajo Android Open Source Project el 9 de julio de 2012.

Por otra parte, el primer dispositivo que funcionaba con esta versión de Android fue el Nexus 7, comercializado el 13 de julio de 2012.

A partir de la versión 4.1 se publicaron dos actualizaciones: 4.1.1, el 23 de julio de 2012, y, 4.1.2, el 9 de octubre de 2012. Las características, divididas por versiones, fueron las siguientes:

4.1

- Interfaz de usuario más fluida:
 - Sincronización Vsync en todos los dibujos y animaciones dentro del marco Android, incluyendo evento, interacción con la pantalla y la pantalla de actualización.
 - Triple buffer en los pipelines gráficos
- Mejoras en accesibilidad.
- Soporte de texto bi-direccional y otros lenguajes.
- Mapas de teclado usuario-instalable.
- Notificaciones expandibles.
- Capacidad para desactivar las notificaciones de forma específica para una aplicación.
- Accesos directos y widgets pueden automáticamente ser reorganizadas o redimensionadas para permitir que nuevos ítems encajen en la pantalla de inicio.
- Transferencia de datos Bluetooth para Android Beam.
- Dictado de voz offline.

- Nuevo diseño de la interfaz para tablets con pantallas pequeñas o phablets (cuyas pantallas se asemejan más a la de un teléfono).
- Mejoras en búsqueda por voz.
- Mejoras en la aplicación de la cámara.
- Google Wallet (para el Nexus 7).
- Fotos de contactos en alta resolución en Google+.
- Aplicación de búsqueda Google Now.
- Audio multicanal.
- Audio USB (Conversor digital-analógico para sonido externo).
- Encadenamiento de audio (también conocido como reproducción sin pausas).
- El navegador de serie de Android es remplazado por la versión móvil de Google Chrome en dispositivos con Android 4.1 preinstalado.
- Habilidad para otros Launchers para añadir widgets desde el cajón de aplicaciones sin necesidad de acceso root.

4.1.1

- Arreglado un fallo en el Nexus 7 con respecto a la incapacidad de cambiar de orientación de la pantalla en cualquier aplicación.

4.1.2

- Soporte de rotación de la pantalla principal.
- Arreglo de fallos y mejoras en rendimiento.
- Notificaciones expansión/contracción con un dedo.

Android 4.2 Jelly Bean (Gummy Bear)

Debido al Huracán Sandy, el evento programado para el 29 de octubre de 2012 en el que se pretendía anunciar la versión 4.2 de Android fue cancelado. En vez de replanificar el evento, Google anunció la nueva versión a través de un comunicado de prensa, con el siguiente eslogan: A new flavor of Jelly Bean. El primer dispositivo en correr Android 4.2 fue el Nexus 4 de LG y el Nexus 10 de Samsung, los cuales fueron lanzados el 13 de noviembre de 2012.

Además de la versión 4.2, se crearon dos actualizaciones más bajo esta denominación: 4.2.1, publicada el 27 de noviembre de 2012, y, 4.2.2, lanzada el 11 de abril de 2013. Las características por versión son las siguientes:

4.2

- Fotos panorámicas con "Photo Sphere".
- Teclado con escritura gestual.
- Mejoras de la pantalla de bloqueo, incluyendo soporte para widgets y la posibilidad de deslizar con el dedo e ir directamente a la cámara.
- Controles en las notificaciones ("Quick Settings"). Protectores de pantalla "Daydream", que muestran información cuando el equipo está inactivo o conectado por USB (docked).
- Múltiples cuentas de usuario (sólo en tablets).
- Soporte para pantallas inalámbricas (Miracast).
- Mejoras en accesibilidad: ampliación a pantalla completa en tres niveles, panorámica y zoom con los dedos. Salida de voz y navegación en modo gestos para usuarios invidentes.
- Nueva aplicación del reloj con función de reloj mundial, cronómetro y temporizador.
- Todos los dispositivos ahora usan el mismo diseño de interfaz, previamente adaptado desde los teléfonos en 4.1 para las tablets pequeñas (con botones centrados en la pantalla, la barra de sistema en la parte superior de la pantalla, y una pantalla de inicio con un dock y el menú de aplicaciones centrado), independientemente del tamaño de la pantalla.
- Incrementado el número de notificaciones extendidas y notificaciones accionables para más aplicaciones, permitiendo responder a ciertas notificaciones con la barra de notificaciones sin lanzar la aplicación directamente.
- SELinux
- Siempre-activa VPN
- Confirmación SMS Premium SMS

4.2.1

- Arreglado un fallo que en la aplicación "People", diciembre no era mostrado cuando se agregaba a un contacto.
- Agregado Bluetooth para gamepads y joysticks como dispositivos HID soportados.

4.2.2

- Arreglados fallos del streaming de audio por Bluetooth.
- Arreglado fallo de cámara que dejaba flash pegado.
- El presionado-largo sobre los iconos de Wi-Fi y Bluetooth en Quick Settings ahora cambia el estado activado / desactivado (on/off).
- Nuevas notificaciones de descarga, que ahora muestran el porcentaje y el tiempo estimado que queda para las descargas activas de aplicaciones.
- Nuevos sonidos para la carga en forma inalámbrica y batería baja.
- Eliminada la opción de mostrar todas las llamadas en la lista de llamadas de la aplicación teléfono.
- Nueva animación de la aplicación galería la cual permite carga más rápida.
- Lista blanca de depuración USB.
- Mejoras en rendimiento y arreglo de fallos menores.

Android 4.3 Jelly Bean

Google lanza la versión Android 4.3 Jelly Bean el 24 de julio de 2013, bajo el lema "An even sweeter Jelly Bean", y, durante el evento "Desayuno con Sundar Pinchai". Esta persona es un hombre de negocios indio, que ocupa el cargo de vicepresidente de la compañía Google. La versión fue entrenada en la nueva generación de los modelos Nexus 7, a partir del 30 de julio del mismo año. Además de esta versión, se publicó una actualización más, llamada 4.3.1, y lanzada el 3 de octubre de 2013. Por versiones, estas son las características del nuevo Android:

4.3

- Soporte para Bluetooth de Baja Energía.
- OpenGL ES 3.0
- Modo de perfiles con acceso restringido.
- DRM APIs de mayor calidad.
- Mejora en la escritura
- Cambio de usuarios más rápida
- Soporte para Hebreo y Árabe
- Locación de WiFi en segundo plano
- Teclado de marcación automática completo.
- Añadido el soporte para 5 idiomas más.
- Opciones para creadores de Apps
- Logs para el desarrollador y mejoras en el análisis.
- Soporte para los sistemas Geofencing y WiFi APIs de escaneo.
- Mejoras en la seguridad
- Ya no es necesario pulsar el icono del micrófono para realizar una búsqueda de voz. Solo hay que decir "OK Google" y en seguida ordenar al equipo lo que se necesite.
- Se incorpora llamada por gestos, al enviar un e-mail.
- Permite enviar a la impresora fotos, documentos y páginas web desde el smartphone o tableta de manera inalámbrica estando conectada a Google Cloud Print o mediante una aplicación predeterminada.
- La nueva aplicación de teléfono prioriza automáticamente los contactos basada en las personas con las que habla la mayoría de las veces.
- Cuando reciba una llamada de un número de teléfono no están en la agenda, el teléfono buscará coincidencias de las empresas con una lista local de Google Maps.
- Nueva máquina virtual de ejecución experimental, ART.
- Se ha re-diseñado la aplicación Descargas.
- Añade soporte completo para Chromecast.
- Optimizado para funcionar tan solo con 512MB de Ram (Project Svelte).

4.3.1

- Corrección de errores y pequeños retoques para el LTE Nexus 7.

Android 4.4 KitKat

Anunciado por Google el 3 de septiembre de 2013, hizo su aparición en el mercado con el dispositivo Nexus 5 el 31 de octubre de 2013. Ha sido optimizado para funcionar en una gama más amplia de dispositivos que las versiones de Android anteriores, teniendo 512 MB de RAM como mínimo recomendado; esas mejoras eran conocidas como "Proyecto Svelte" internamente en Google. Kit Kat no fue su nombre original, el cual iba a ser "Key Lime Pie", pero fue cambiado debido a que "muy pocas personas saben realmente el sabor de una tarta de lima".

Además de la versión inicial 4.4, han sido publicadas las siguientes actualizaciones: 4.4.1, el 5 de diciembre de 2013, 4.4.2, el 9 de diciembre de 2013, 4.4.3, el 2 de junio de 2014, y, por último, la 4.4.4, lanzada el 19 de junio de 2014. Por versiones, estas son las características de los últimos lanzamientos de Android:

4.4

- Se corrige un error que había con el widget reproductor que aparecía en la pantalla de bloqueo cuando estábamos reproduciendo contenido multimedia con una aplicación compatible. Al mantener pulsado el botón pausa nos permite retroceder o avanzar la canción o vídeo, pero en Android 4.4 no funcionaba correctamente y nos podía cambiar de pantalla de bloqueo.
- En Android 4.3 se añadía soporte a Bluetooth Smart (Low Energy) pero sólo permitía sincronizar hasta 4 dispositivos. A partir de Android 4.4.1 el límite se aumenta hasta 7 dispositivos. También se mejora el control de volumen único para dispositivos con Bluetooth.
- Otra mejora el rendimiento del sistema, corrigen los fallos con la sincronización de cuentas de correo Exchange, la alineación de los iconos de la barra de estado, problemas de volumen con las aplicaciones y mejora el runtime ART para que funcione correctamente con más aplicaciones como Whatsapp.

- El widget de acceso rápido a Ajustes que hasta ahora permitía cambiar entre activar y desactivar la localización, ahora se puede configurar también para cambiar los modos de ahorro de energía.
- Se implementó el acceso directo a Fotos desde Cámara, siendo éste un paso más hacia la integración completa como galería por defecto.
- Se arregló el bug que impedía que la barra fuese translúcida en la pantalla de bloqueo, así como se mejoró la alineación de los iconos de conexiones, cobertura, batería y reloj.
- Se puede ocultar este teclado simplemente pulsando en una parte vacía de la pantalla. Una pequeña muestra de cómo van puliendo la interfaz.

4.4.1

- En Android 4.4 KitKat existía un bug que hacía que algunas apps que incluyen sonido sonasen más alto que otras por el altavoz, aunque todas ellas tengan el volumen al máximo. Este error se ha corregido y ahora todas alcanzan el máximo volumen permitido.
- Mejoras en el enfoque automático, balance de blancos y HDR+ para la cámara del Nexus 5.
- Mejor compatibilidad de aplicaciones para el Android Runtime.
- Aplicación de la cámara para cargar las fotos directamente en Google+.
- Varias correcciones de bugs.

4.4.2

- Se ha arreglado un fallo que simulaba la opción “elegir siempre” cuando aparecía la opción de compartir con distintas aplicaciones, escogiéndose así aplicaciones favoritas sin que el usuario lo hiciese.
- Se ha solucionado el fallo de seguridad que afectaba a los SMS de tipo flash, que permitía mediante el uso de ese tipo de mensajes bloquear, reiniciar e incluso dejar al terminal sin conexión de red.
- Se ha arreglado un fallo que impedía mantener pulsado el botón de Pause en el widget del reproductor de música en el lockscreen (o cualquier otro reproductor que soportara esta acción) para que apareciese una barra de estado de la canción.

- Otra de las novedades de 4.4.2 es que App Ops, la pantalla desde la que gestionar los permisos y notificaciones de las aplicaciones que hemos instalado, ha desaparecido.
- Se ha arreglado un problema de batería que provoca el gasto de esta muy rápidamente. Quien presentó el problema fue el fabricante Motorola, los cuales fueron los únicos en recibir esta corrección.

4.4.3

En la actualización 4.4.3, Android se centra principalmente en corregir el sistema operativo (bugs), aunque termina incluyendo una serie de mejoras con respecto a la versión anterior:

- Arreglos en la conexión de datos
- Optimización del servicio mm-qcamera-daemon
- Arreglos de enfoque de cámara en los modos HDR y normal
- Arreglos de 'wakelock' en el gestor de batería
- Múltiples correcciones en el soporte Bluetooth
- Solución de reinicios aleatorios
- Solución de la desaparición de accesos directos de algunas apps tras su actualización
- Arreglos de seguridad en la depuración USB
- Arreglos de seguridad en los accesos directos de las apps
- Solución en la conexión automática Wi-Fi
- Ajustes en MMS, Email/Exchange, Calendario, Contactos, DSP, IPv6 y VPN
- Solución del atasco en la pantalla de activación
- Arreglo del LED en las llamadas perdidas
- Arreglo del gráfico de uso de datos
- Arreglos en VoIP
- Corrección para conformidad de la FCC
- Nueva Interfaz del marcador.
- Corrección de subtítulos

4.4.4

- CVE-2014-0224 fija, eliminando una vulnerabilidad OpenSSL

Android 5.0 L

El pasado 25 de junio Sundar Pichai, vicepresidente senior de Google, concedió una entrevista al portal Businessweek, en la que desveló que la nueva versión de Android, conocida por “L” saldrá en otoño de 2014.

Se espera que Android “L” traiga consigo cambios importantes con respecto a versiones anteriores, como una nueva interfaz o compatibilidad con procesadores de 64 bits.

Según otras fuentes, y es posible que Google ya lleve un tiempo probando el Android 5.0 en dispositivos Nexus desde hace unos meses. En principio, esta actualización debutaría en la colección de dispositivos Nexus: 4, 5, 7 y 10, quedando el Galaxy Nexus sin actualización por llevar al menos dos años ya en el mercado.

Otro de los detalles que también están en duda, es el nombre de la nueva versión. Algunas fuentes extienden la “L” de su nombre a “Lemon Drop Sunshine”.

2.1.3 Android en el mercado

Partiendo de los datos recogidos en la primera versión de este proyecto, he aquí una actualización de los datos a partir de la última fecha recogida en el mismo: septiembre 2013. Desde entonces, el mercado para Android ha sufrido estos importantes cambios:

- En noviembre de 2013, Andy Rubin, co-fundador y ex CEO de Danger Inc. y Android, comentó que se activaban 1.500.000 dispositivos diariamente.
- En julio de 2014, este era la cuota de versiones de Android:

| Versión | Nombre en código | Fecha de distribución | <u>API</u> level | Cuota (7 de Julio, 2014) |
|------------------------------------|---|------------------------------|-------------------------|---------------------------------|
| <u>4.4</u> | <u>Kit Kat</u> | 31 de octubre de 2013 | 19 | 17.9% |
| <u>4.3</u> | <u>Jelly Bean</u> | 24 de julio de 2013 | 18 | 9.0% |
| <u>4.2.x</u> | <u>Jelly Bean</u> | 13 de noviembre de 2012 | 17 | 19.7% |
| <u>4.1.x</u> | <u>Jelly Bean</u> | 9 de julio de 2012 | 16 | 27.8% |
| <u>4.0.x</u> | <u>Ice Cream Sandwich</u> | 16 de diciembre de 2011 | 15 | 11.4% |
| <u>3.2</u> | <u>Honeycomb</u> | 15 de julio de 2011 | 13 | <0.1% |
| <u>2.3.3–2.3.7</u> | <u>Gingerbread</u> | 9 de febrero de 2011 | 10 | 13.5% |
| <u>2.2</u> | <u>Froyo</u> | 20 de mayo de 2010 | 8 | 0.7% |

Figura2.2: Cuota de mercado actual (fuente: Xatakaandroid.com) - Wikipedia

Con lo que respecta a España, en mayo de 2014 se publicó un estudio en el que, en marzo del mismo año, Android perdía por primera vez cuota de mercado en España frente a sistemas operativos como iOS y Windows Phone. Durante el mes siguiente, se repetía la caída. El autor de este estudio fue la consultora Kantar Worldpanel ComTech.

Los datos exactos de dicha pérdida pueden consultarse detalladamente a continuación. A modo de resumen, con respecto al trimestre analizado para este estudio, Android ha experimentado un descenso de ventas de un 5,9%, pasando del 92% al 86,1% de cuota de mercado con respecto al mismo periodo del año anterior. Solo en el último mes de este estudio, abril 2014, el porcentaje ha descendido un 2,5%. A pesar de estos datos, España sigue siendo el país donde Android tiene más cuota de mercado, seguido por China, con un 79,8%.

Además de los datos correspondientes al mercado nacional, también pueden consultarse otros países líderes en uso de nuevas tecnologías:

Smartphone OS Sales Share (%)

| Germany | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change | USA | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change |
|------------|----------------|----------------|--------------|------------|----------------|----------------|--------------|
| Android | 74.9 | 78.5 | 3.6 | Android | 51.8 | 59.1 | 7.3 |
| BlackBerry | 0.4 | 0.2 | -0.2 | BlackBerry | 0.7 | 1.0 | 0.3 |
| iOS | 15.7 | 14.1 | -1.6 | iOS | 41.4 | 34.6 | -6.8 |
| Windows | 6.6 | 6.9 | 0.3 | Windows | 5.6 | 4.7 | -0.9 |
| Other | 2.4 | 0.3 | -2.1 | Other | 0.6 | 0.6 | 0.0 |
| GB | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change | China | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change |
| Android | 57.2 | 58.2 | 1.0 | Android | 71.9 | 79.8 | 7.9 |
| BlackBerry | 5.5 | 1.6 | -3.9 | BlackBerry | 0.3 | 0.0 | -0.3 |
| iOS | 28.0 | 30.2 | 2.2 | iOS | 23.7 | 17.5 | -6.2 |
| Windows | 8.6 | 9.5 | 0.9 | Windows | 2.2 | 0.8 | -1.4 |
| Other | 0.6 | 0.5 | -0.1 | Other | 1.9 | 1.9 | 0.0 |
| France | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change | Australia | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change |
| Android | 68.0 | 66.6 | -1.4 | Android | 64.1 | 60.8 | -3.3 |
| BlackBerry | 3.8 | 1.1 | -2.7 | BlackBerry | 0.6 | 0.9 | 0.3 |
| iOS | 16.9 | 20.2 | 3.3 | iOS | 28.1 | 30.1 | 2.0 |
| Windows | 6.2 | 10.1 | 3.9 | Windows | 5.0 | 6.7 | 1.7 |
| Other | 5.1 | 2.0 | -3.1 | Other | 2.2 | 1.5 | -0.7 |
| Italy | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change | Japan | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change |
| Android | 66.7 | 72.5 | 5.8 | Android | 44.4 | 43.6 | -0.8 |
| BlackBerry | 2.0 | 1.0 | -1.0 | BlackBerry | 0.8 | 0.0 | -0.8 |
| iOS | 16.5 | 13.4 | -3.1 | iOS | 49.8 | 55.4 | 5.6 |
| Windows | 10.5 | 11.8 | 1.3 | Windows | 0.6 | 1.0 | 0.4 |
| Other | 4.3 | 1.2 | -3.1 | Other | 4.4 | 0.0 | -4.4 |
| Spain | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change | EU5 | 3 m/e Apr 2013 | 3 m/e Apr 2014 | % pt. Change |
| Android | 92.0 | 86.1 | -5.9 | Android | 70.8 | 72.4 | 1.7 |
| BlackBerry | 0.1 | 0.0 | -0.1 | BlackBerry | 2.4 | 0.8 | -1.7 |
| iOS | 4.8 | 8.7 | 3.9 | iOS | 17.2 | 17.5 | 0.2 |
| Windows | 1.7 | 4.6 | 2.9 | Windows | 6.9 | 8.4 | 1.6 |
| Other | 1.4 | 0.6 | -0.8 | Other | 2.7 | 0.9 | -1.8 |

Figura2.3: Comparativa de uso de Android en diferentes países, con respecto a otros sistemas.

2.2.1 Arquitectura

Android es un sistema diseñado por capas. Su núcleo está formado por el sistema operativo Linux 2.6. Gracias a esta característica, se accede a la parte hardware de los dispositivos a la par que le permite ser compatible con muchos de los drivers creados para Linux. Proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso o la pila de protocolos. En resumen, es una abstracción entre el hardware y el resto de la pila.

En la siguiente imagen se puede ver la arquitectura descrita:

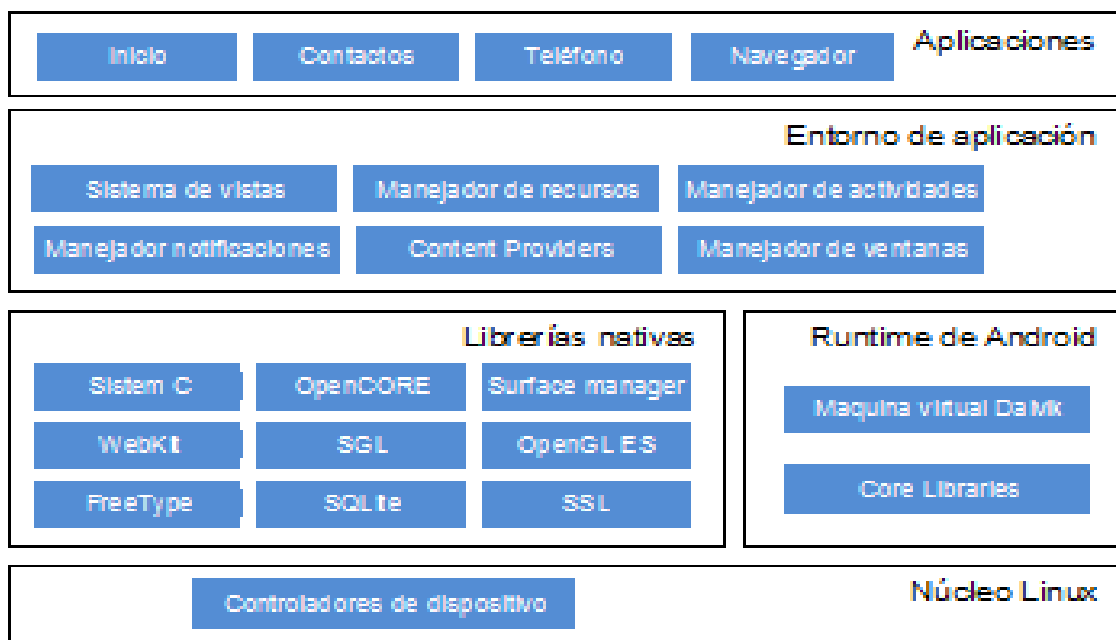


Figura2.4: Arquitectura del sistema operativo Android

- **Aplicaciones**

Este grupo está formado por el conjunto de aplicaciones instaladas en nuestro dispositivo con sistema operativo Android. Todas las aplicaciones han de ejecutarse utilizando la máquina virtual Dalvik para garantizar la seguridad del sistema. Generalmente, las aplicaciones para Android están escritas en lenguaje de programación Java, utilizando el SDK (software development kit) de Android proporciona, aunque también existe la posibilidad de desarrollar aplicaciones utilizando C/C++, utilizando para ello el Android NDK (native development kit).

Las aplicaciones en Android pueden estar compuestas por cinco bloques, sin necesidad de utilizarlos todos en cada aplicación desarrollada: Activity, Intent, Broadcast, Services y Content Providers. Por otra parte, cada dispositivo Android, ya sea Tablet PC o un teléfono inteligente, traen instaladas por defecto una serie de ellas, como pueden ser el cliente de correo, navegadores web, mapas, gestor de contactos, acceso al Play Store o búsqueda mediante Google.

- **Framework**

El marco de aplicaciones da acceso completo a los programadores a las mismas APIs utilizadas por las aplicaciones básicas. Cualquier desarrollador puede reutilizar estos componentes de forma sencilla, teniendo claro que las normas de seguridad impuestas por el framework no se van a ver en peligro.

Teniendo en cuenta que esta capa está diseñada para simplificar la reutilización de componentes, algunas aplicaciones publican sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad impuestas por el framework). Este mecanismo permite al usuario reemplazar componentes.

Otro de los puntos fuertes del entorno de aplicación de Android es la utilización del lenguaje de programación Java. El SDK de Android no está a la altura del mismo de Java, ya que no puede ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE). Aun así, es compatible en una amplia parte.

Los servicios más importantes que incluye son:

- **Views:** extenso conjunto de vistas, (parte visual de los componentes).
- **Resource Manager:** proporciona acceso a recursos que no son en código.
- **Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- **Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

- **Librerías**

Estas librerías, desarrolladas en C/C++, usadas en varios componentes Android, están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- **System C library:** una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework:** librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.

- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- **SGL:** motor de gráficos 2D.
- **Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite:** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** proporciona servicios de encriptación Secure Socket Layer.

- **Android Runtime**

Siguiendo el modelo de máquina virtual Java, Android creó su propia máquina virtual, llamada Dalvik. Las razones fueron claras: los dispositivos para los que está pensando Android tienen poca memoria y un procesador limitado, de ahí que una máquina virtual Java estándar no pueda utilizarse. Más adelante se hablará en profundidad acerca de Dalvik.

Completando el Runtime se encuentran las Core Libraries, con la mayoría de las librerías disponibles en el lenguaje Java.

- **Kernel de Linux**

Tal y como se ha comentado al inicio de la explicación acerca de la arquitectura Android, el núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el multiproceso, el soporte de drivers para dispositivos, la pila de protocolos y el manejo de la memoria. Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila, es decir, es la única que depende del hardware.

Varias fueron las razones que llevaron a tomar la decisión del uso de la versión 2.6 del kernel de Linux para utilizarlo como núcleo de Android:

- Ser un sistema operativo de código abierto
- Respuesta más ágil que sus predecesores
- El kernel puede ser interrumpido (solo en ciertos casos)
- Renovación del kernel en cuanto al manejo de hilos.

2.2.2 Dalvik

En la actualidad, los requisitos hardware en los dispositivos móviles con sistema operativo Android suelen ser bastante limitados. Se espera que en un futuro puedan equipararse a los de cualquier PC actual, pero hasta entonces los recursos son los que son. Debido a esto, y a las necesidades que requiere la ejecución de aplicaciones sobre la máquina virtual Java, se tomó la decisión de crear una nueva máquina virtual, que pudiera trabajar con recursos más limitados y así poder instalarla en cada dispositivo Android. El resultado se llamó Dalvik.

Diseñada por Dan Bornstein, con contribuciones de otros ingenieros de Google, permite ejecutar las aplicaciones desarrolladas en Java para Android. Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como gestión de hilos y el manejo de la memoria a bajo nivel.

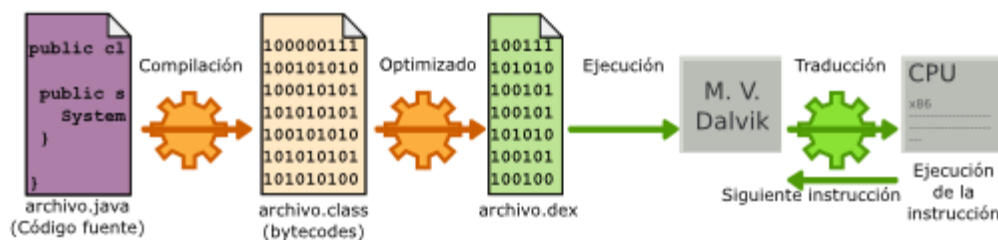


Figura 2.5: Mecanismo de compilación y traducción de una instrucción

Por último, durante la ejecución de varias instancias simultáneamente, el rendimiento de la memoria del dispositivo debe verse afectado en la menor medida de lo posible. Este hecho lo ha logrado Dalvik a la perfección, de forma que las aplicaciones se ven protegidas para los casos que una de ellas se cierre o cometa un fallo, no afecte a la ejecución del resto.

- **Diferencia con la Máquina Virtual de Java**

El lenguaje de programación Java ha sido el principal causante de la revolución tecnológica de los últimos veinte años. Sin su existencia, Internet no sería conocido como lo es hoy, o al menos, no estaría tan extendido. Su funcionamiento se basa en ejecutar su código sobre la máquina virtual Java. Una vez compilado el código, la máquina virtual Java (JVM) es capaz de interpretar y convertir en código nativo que el sistema operativo es capaz de interpretar.

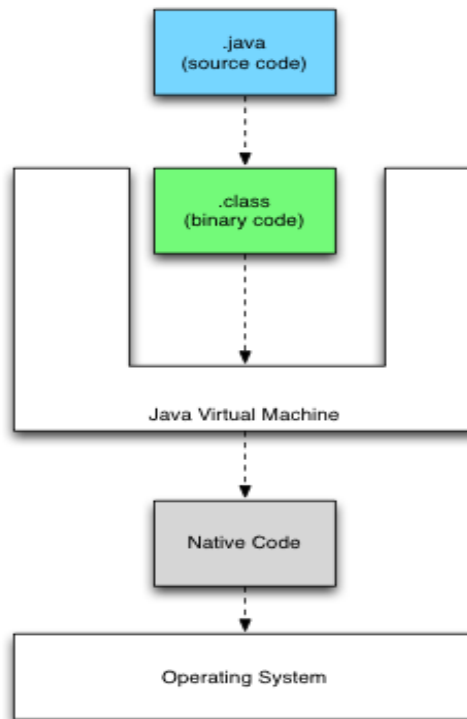


Figura 2.6: Mecanismo de compilación y traducción de una instrucción - JVM

Aunque se utiliza el lenguaje Java para programar las aplicaciones Android, el bytecode de Java no es ejecutable en un sistema Android. Sin embargo, la herramienta dx incluida en el SDK de Android permite transformar los archivos Class de Java compilados por un compilador Java al formato de archivos Dex.

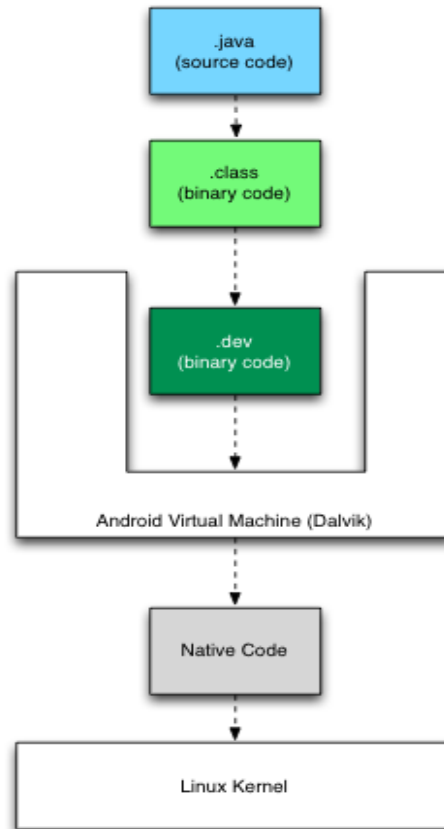


Figura 2.7: Mecanismo de compilación y traducción de una instrucción – Dalvik

Al igual que para el caso anterior, las librerías Java que se utilizan para desarrollar aplicaciones en Android varían con respecto a las utilizadas en Java Standard Edition (Java SE) o en Java Mobile Edition (Java ME).

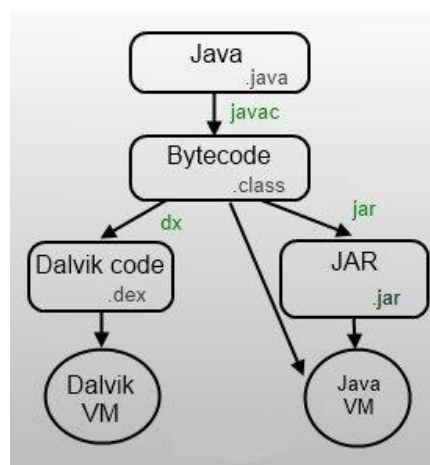


Figura 2.8: Comparativa JVM Vs Dalvik

- **Dalvik y la optimización en aplicaciones Android**

Uno de los principales objetivos del uso de Dalvik era la reducción del tamaño del programa. Dalvik lo ha conseguido, de la siguiente forma: buscando información duplicada en las diversas clases y reutilizándola.

Otro de los objetivos era tratar de imitar, dentro de las posibilidades, los mejores componentes de la máquina virtual Java. Uno de ellos, el conocido como “Recolector de basura”, que libera el espacio en memoria de objetos que ya no son utilizados en nuestros programas, ha sido mejorado en Android con el fin de mantener siempre libre la máxima memoria posible. Teniendo en cuenta lo comentado con anterioridad sobre las limitaciones acerca de los dispositivos Android actuales, este punto es muy importante.

De igual forma, el hecho de que Android haga un uso extenso del lenguaje XML para definir las interfaces gráficas y otros elementos, implica que estos archivos deben ser linkados a la hora de compilar y para que su conversión a bytecode pueda mejorar el rendimiento de las aplicaciones.

2.2.3 Componentes

El sistema operativo Android ofrece una serie de componentes a utilizar en las aplicaciones desarrolladas para el mismo. Éstos componentes son: Activity, Service, Intent, Broadcast receiver y Content provider. El uso de los cinco componentes no es obligatorio en cada aplicación, pudiendo utilizarse libremente en función de las necesidades del desarrollo. Los utilizados deben aparecer de forma explícita en el fichero AndroidManifest.xml, donde se encuentran definidos otros datos importantes como los permisos, siendo un fichero básico en cualquier aplicación. A continuación, se describirán los cinco componentes:

- **Activity**

Una actividad en Android consiste en un conjunto de elementos básicos de visualización, que suelen ser las pantallas de la aplicación. Cada una de estas pantallas es representada por una actividad (Activity). Su función principal es la creación del interfaz de usuario. Habitualmente suelen ser una colección de actividades las que forman una interfaz de usuario. De naturaleza independiente, las actividades en Android trabajan por un objetivo común. Toda actividad ha de pertenecer a una clase descendiente de Activity.

En la gran mayoría de aplicaciones, las actividades son múltiples, es decir, cada aplicación tendrá varias pantallas a las que el usuario podrá acceder. El hecho de pasar de una pantalla a otra se consigue mediante la inicialización de una nueva actividad. En estos casos, la pantalla anterior, quedará almacenada en una pila, de forma que el usuario podrá acceder a ella cuando desee, permitiendo navegar entre actividades abiertas previamente.

- **Intent**

El concepto de Intent se encuentra estrechamente ligado con la inicialización de las actividades. Además, se puede definir como la intención de realizar una acción (en este caso, comenzar una nueva actividad). Un Intent representa la voluntad de realizar alguna acción, como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que se requiere lanzar un Activity, un Service, un anuncio de tipo broadcast o una comunicación con un Service. Los componentes lanzados pueden ser internos o externos a nuestra aplicación. También se utilizan los Intents para el intercambio de información entre estos componentes.

- **Service**

Un servicio es un proceso que se ejecuta en segundo plano, sin la necesidad de una interacción con el usuario. Es algo parecido a un demonio en Unix o a un servicio en Windows.

En Android se dispone de dos tipos de servicios: servicios locales, que son ejecutados en el mismo proceso y servicios remotos, que son ejecutados en procesos separados.

El ejemplo más común que se da para comprender el concepto de servicio es el reproductor de música que viene incluido en el sistema operativo. Esta aplicación está dotada de una interfaz gráfica (o Activity) donde se permite al usuario elegir entre una lista de canciones, generar listas de reproducción así como el resto de acciones típicas de estos reproductores. Sin embargo, en el momento en que el usuario comienza a reproducir un archivo de audio, puede seguir navegando entre las canciones e incluso comenzar a utilizar otras aplicaciones. Esto se debe a que la reproducción se realiza mediante un servicio y se ejecuta en background. De igual manera que las actividades, se implementa mediante la clase con su mismo nombre.

- **Broadcast receiver**

Un broadcast receiver es un componente destinado a realizar acciones cuando se producen anuncios globales de tipo broadcast. Existen muchos anuncios originados por el sistema, como por ejemplo Batería baja o llamada entrante. Las aplicaciones también pueden lanzar un anuncio broadcast o incluso crear nuevos tipos. Los broadcast receiver permiten crear aplicaciones mucho más integradas en el entorno donde se ejecutan. También es posible, mediante el uso de los intents, generar mensajes broadcast en la propia aplicación que estén dirigidos a cualquier Broadcast receiver que se encuentre activo.

Este componente tampoco tiene una interfaz gráfica asociada, pero pueden utilizar el API Notification Manager para avisar de manera no intrusiva al usuario del evento que se ha producido a través de la barra de notificaciones presente en el sistema. Al igual que los componentes anteriores, los broadcast receivers, se implementan a través de la clase con su mismo nombre.

- **Content provider**

Un terminal Android puede necesitar, en muchas ocasiones, que las aplicaciones instaladas compartan información. Para ello, Android define un mecanismo estándar destinado a que las aplicaciones compartan datos sin posibilidades de poner en riesgo la seguridad del sistema de ficheros. A través de esta herramienta, se puede acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación.

Algunos de estos Content providers ya existen implementados y permiten compartir todo tipo de datos, información de los contactos, imágenes, video, mensajes de texto o incluso audio. `Android.provider`, además de ser el lugar donde se encuentran estos Content providers, ofrece la posibilidad de crear nuevos propios.

Los Content providers, son objetos de la clase `ContentProvider`, que se encuentra localizada en el paquete `android.content`. Una URI se encarga de la identificación de cada objeto de esta clase y, a través de la misma, el resto de aplicaciones deben acceder a él. A la hora de crear un nuevo Content provider para una aplicación se debe tener en cuenta que los datos que se vayan a hacer públicos se almacenarán habitualmente como una tabla de una base de datos SQLite donde cada columna se refiere a un tipo de dato y cada fila representa un registro.

2.2.4 Ciclo de vida de una aplicación

Un usuario acostumbrado a utilizar sistemas operativos para PCs, como por ejemplo Windows, sabe que el ciclo de vida de una aplicación está en su mano. Es decir, cuando el usuario desea, puede terminar la ejecución de esa aplicación, y si no lo hace, la aplicación continúa ejecutándose. En Android es bastante diferente a otros sistemas operativos, ya que, el ciclo de vida de una aplicación es controlado principalmente por el sistema, en vez de ser controlado directamente por el usuario. Cada aplicación Android es ejecutada en su propio proceso y es el propio sistema operativo el encargado de lanzar y parar estos procesos, gestionar

su ejecución y decidir qué hacer en función de los recursos disponibles y de las órdenes dadas por el usuario.

En un dispositivo Android, el sistema es quien decide el número de procesos posibles, en función de los recursos de los que dispone. El usuario desconoce este comportamiento, siendo consciente únicamente del paso de una aplicación a otra con una pequeña interacción con el dispositivo. No debe preocuparse sobre cuál es la aplicación que realmente está activa, cuánta memoria está consumiendo, ni si existen o no recursos suficientes para abrir una aplicación adicional.

Otra característica importante del ciclo de vida de una aplicación en Android es el momento de eliminarse el proceso de una aplicación. Si el usuario decide volver a utilizar esa aplicación, se creará de nuevo el proceso, pero el estado que tenía la aplicación en la ejecución anterior se habrá perdido. Para estos casos, es responsabilidad del desarrollador de la aplicación el almacenar el estado de las actividades, siempre que se desee mantener el estado previo una vez reiniciada la aplicación.

2.2.5 Ciclo de vida del componente Activity

Existen cuatro estados dentro del ciclo de vida de una Activity o actividad:

- **Activo** (*Resumed*): La actividad se encuentra en primer plano (Encima de la pila de tareas) e interactuando con el usuario.
- **Pausado** (*Paused*): La actividad sigue siendo visible para el usuario, pero ha perdido el foco. Por ejemplo que se haya mostrado un cuadro de dialogo delante de la actividad. Se debe guardar el estado de la interfaz y los datos de esta actividad antes de entrar en este estado, ya que se podrían perder si el sistema necesita más recursos de memoria.
- **Parado** (*Stopped*): La actividad no es visible para el usuario, queda a disposición del sistema para borrarla de la pila en caso de necesitar memoria.

En cada ocasión en la que se cambia el estado de una actividad, se generarán eventos que podrán ser capturados por ciertos métodos de la actividad. A continuación se muestra un esquema que ilustra esta serie de métodos:

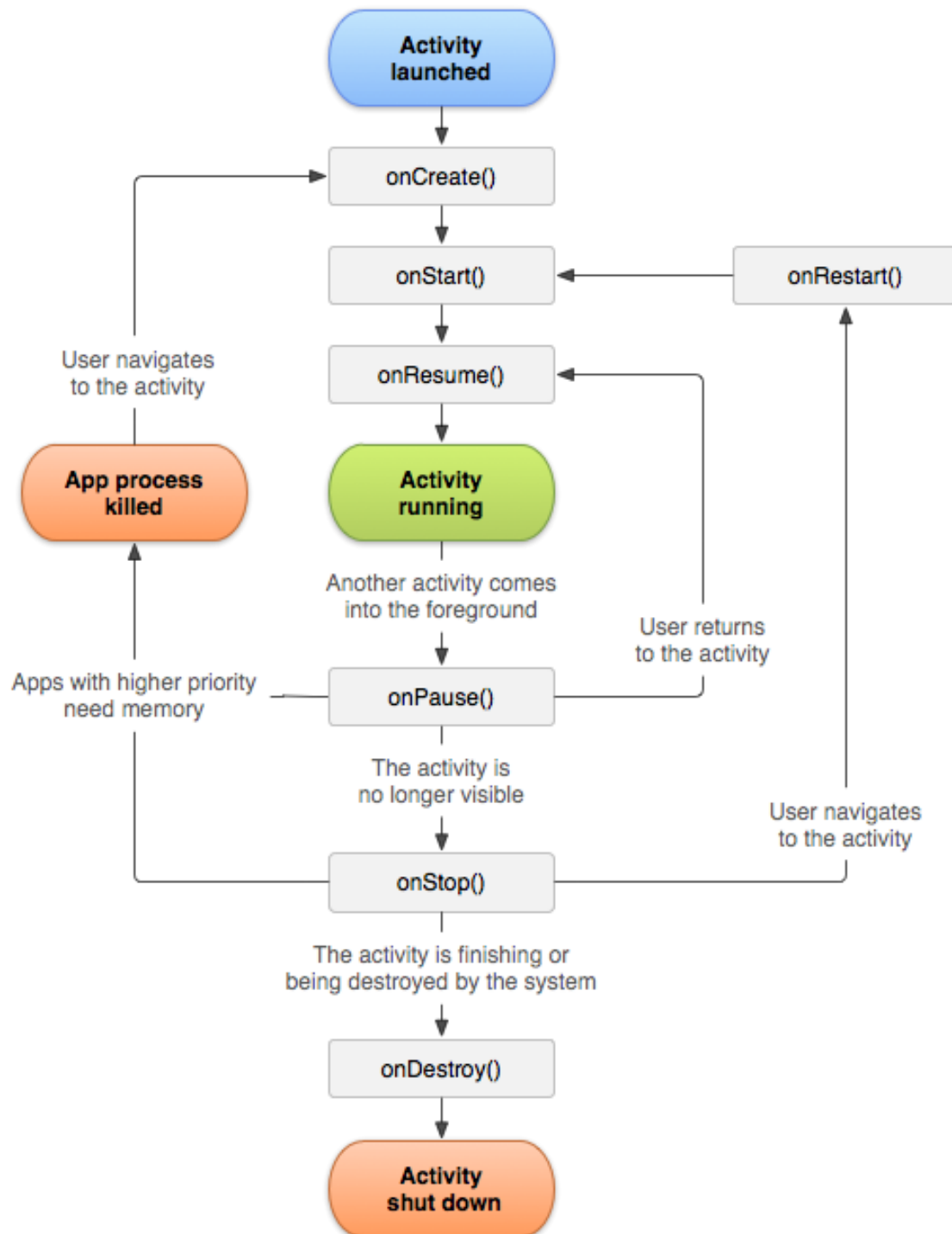


Figura 2.9: Ciclo de vida de un componente Activity

Esta es la descripción de cada uno de los eventos:

- **onCreate(Bundle):** Invocado durante la creación de la actividad, su utilidad consiste en realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Tiene la posibilidad de recibir información de estado de la actividad para el caso de reanudación desde una actividad que ha sido destruida y vuelta a crear.
- **onStart():** Indica que la actividad está a punto de ser mostrada al usuario.
- **onResume():** Invocada en el momento en el que la actividad va a comenzar a interactuar con el usuario, es el lugar idóneo para ejecutar las animaciones y la música.
- **onPause():** Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra actividad es lanzada. Es el momento adecuado para detener animaciones, música o almacenar los datos que estaban en edición.
- **onStop():** La actividad ya no va a ser visible para el usuario. En los casos en los que el sistema disponga de poca memoria, existe la posibilidad de destruirse llamar a este método.
- **onRestart():** Indica que la actividad va a volver a ser representada después de haber pasado por *onStop()*.
- **onDestroy():** Invocada antes de que la actividad sea totalmente destruida, al igual que *onStop*, si el sistema disponga de poca memoria, existe la posibilidad de destruirse llamar a este método. Un ejemplo de su ejecución es, cuando el usuario pulsa el botón <volver> o cuando se llama al método *finish()*.

2.2.6 Ciclo de vida de un Service

Existen dos tipos de Service en Android, en función de la manera en la que hayan sido lanzados. Son los siguientes:

- La primera función permite indicar al sistema que el elemento que se está creando ha de ejecutarse en segundo plano, normalmente durante un largo período de tiempo. Este tipo de servicios son iniciados mediante el método `startService()`, que indica al sistema que lo ejecute de forma indefinida hasta que alguien le indique lo contrario.
- La segunda función permite que nuestra aplicación se comunique con otras aplicaciones, para lo cual ofreceremos ciertas funciones que podrán ser llamada desde otras aplicaciones. Este tipo de servicios son iniciados mediante el método `bindService()`, que permite establecer una conexión con el servicio e invocar alguno de los métodos que son ofrecidos.

A continuación se muestra el diagrama con el ciclo de vida de los servicios:

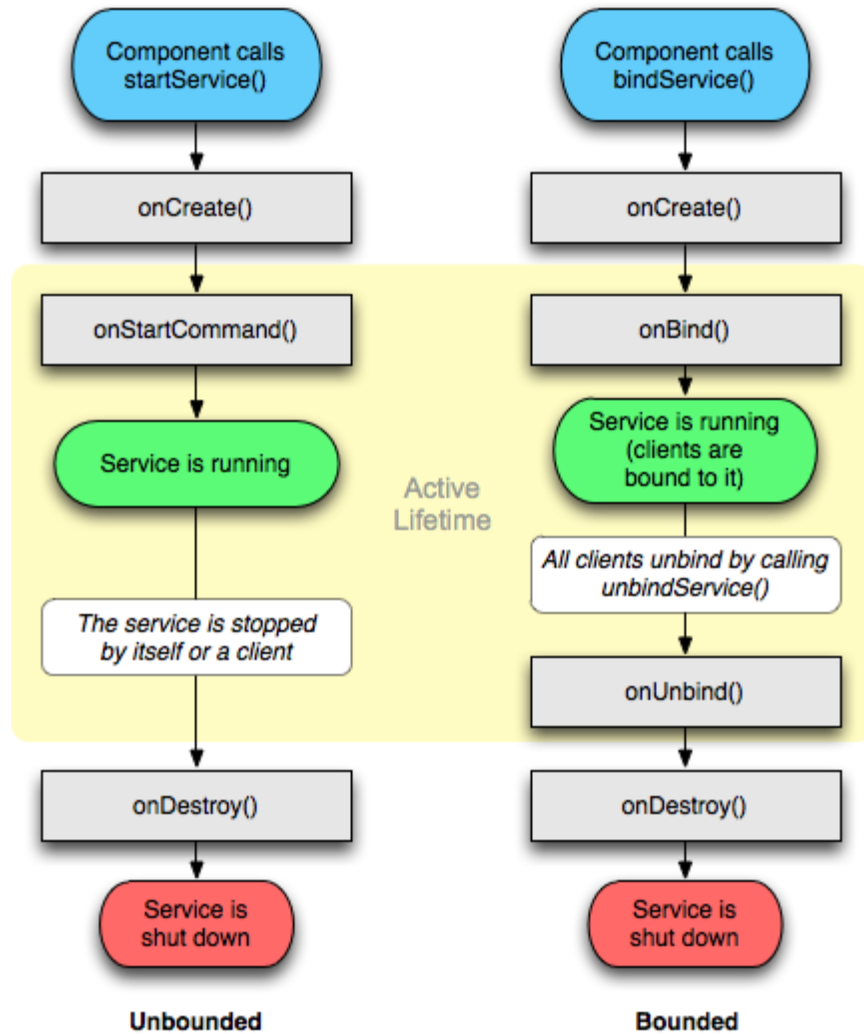


Figura 2.10: Ciclo de vida de un Service

Capítulo 3: Sphero

En este capítulo se hablará de Sphero, nombre comercial del juguete a manejar de forma teledirigida desde el dispositivo Android. Teniendo en cuenta que se ha utilizado la misma versión del juguete, la información aquí aparecida es muy similar a la de la primera versión del proyecto, completándose con las características de la nueva versión disponible en el mercado.

3.1 Introducción a Sphero

Sphero es un dispositivo robótico creado por la compañía Orbotix en 2012. Desde su creación, se han puesto en el mercado dos versiones de este dispositivo, siendo la última de agosto de 2013. Esta bola posee un hardware controlable mediante software (modificación de parámetros en los sensores, límite de velocidad del motor, frecuencia de recepción bluetooth, etc.), además de controlar los movimientos de la bola.

El fabricante de Sphero, la empresa Orbotix, ha desarrollado una serie de aplicaciones propietarias y pone a disposición del usuario un SDK, además de todo un apoyo técnico a través de su web, donde cualquier persona puede acceder a consultar las novedades técnicas así como la información relacionada con el SDK.

El Sphero se controla mediante un dispositivo Tablet PC o un teléfono inteligente, vía Bluetooth. Tanto la Tablet PC como el teléfono inteligente pueden tener sistemas operativos Android o iOS.

Para su funcionamiento, utiliza una batería interna, la cual tiene una duración aproximada de una hora de funcionamiento continuo. Para su recarga, el fabricante provee un sistema de recarga por inducción.

3.2 Características físicas de Sphero

Estas son las principales características del juguete robótico Sphero:

- Tiene un acelerómetro similar al que pueda tener cualquier dispositivo Tablet PC o teléfono inteligente, apoyándose en sus tres ejes de coordenadas: X, Y, Z.

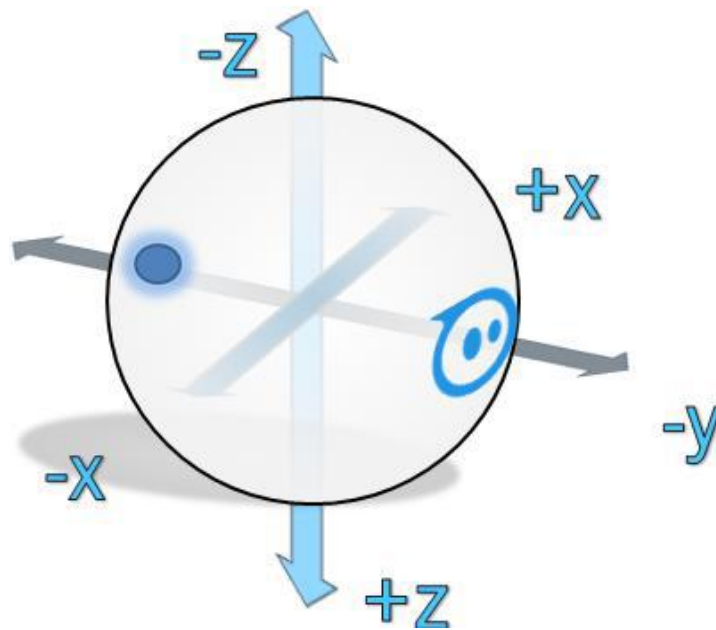
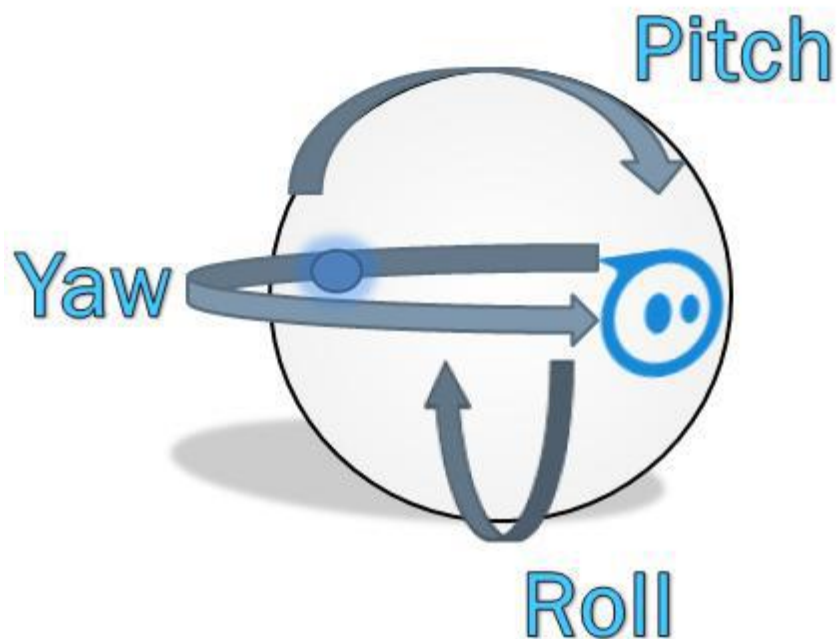


Figura 3.1: Representación de los ejes de dirección

- Dispone de un giroscopio, el cual permite colocar un Sphero en la posición que el usuario desee. Esta posición determinará la dirección de desplazamiento de la bola.



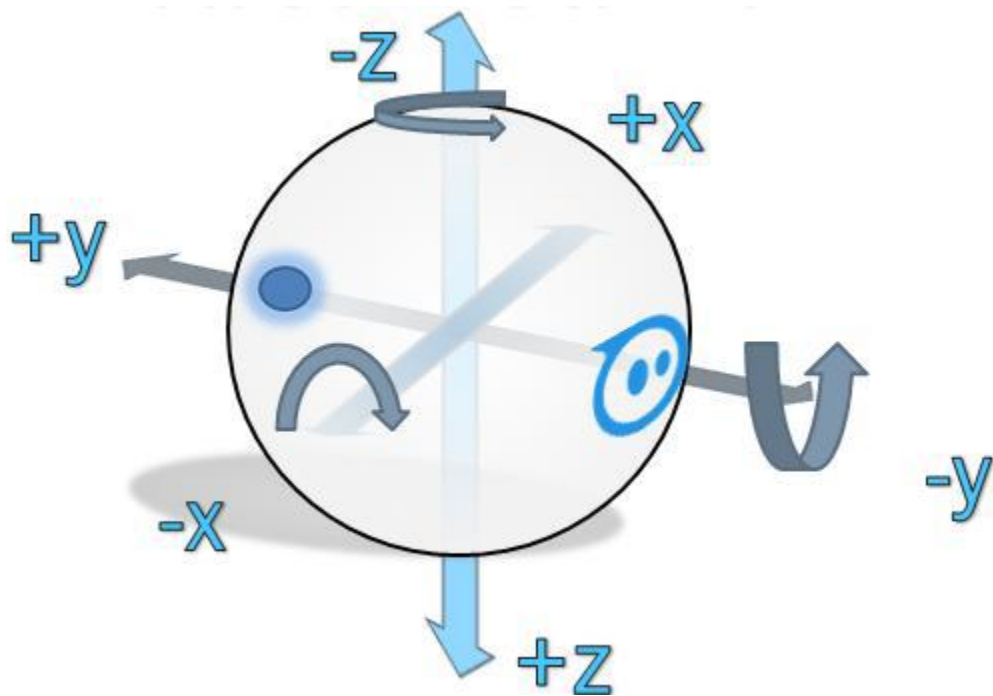


Figura 3.2: Representación del mecanismo de giro del giroscopio

- Posee un LED, al cual el fabricante llama *LEDTail* o LED de la Cola, el cual dará información al usuario acerca del punto en el que se encuentra el giroscopio para realizar movimientos. Este aspecto es de vital importancia, debido a que el control de los movimientos para conocer la dirección en la que va a girar en el avance.

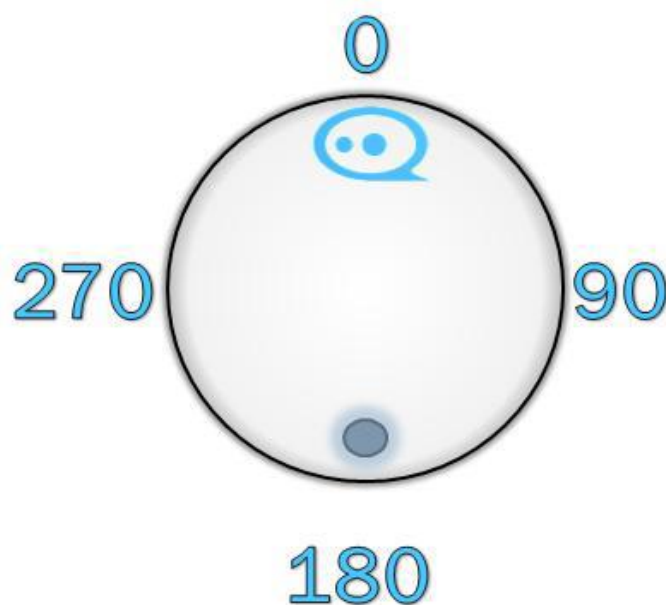


Figura 3.3: Representación de las direcciones en grados respecto a la cabeza

- Incluye un sensor anticolisiones, el cual informa al usuario de las colisiones que se van produciendo.
- Posee un magnetómetro, utilizado para la recarga de la batería mediante inducción electromagnética. Esta característica también anula cualquier instrucción recibida por el Sphero.
- Tiene un sensor de localización, mediante el cual, se puede conocer su posición en todo momento, enviando estos datos al dispositivo que lo está controlando.
- En su interior, una serie de LEDs son capaces de proyectar más de 16 millones de colores combinándose.

3.3 Motivos que llevaron a elegir Sphero

El principal motivo por el cual se ha elegido el dispositivo robótico Sphero era mantener la continuidad de la primera versión de este proyecto. De esta forma, además de poder mejorar algunos de los desarrollos realizados para la primera versión, se podía ahondar en otras características no abordadas para la primera versión. Con esta situación, tenemos un producto mucho más completo, tratando de abarcar, de alguna forma, todas las posibilidades técnicas que ofrece este dispositivo robótico.

Por otra parte, una vez planteado el proyecto y analizadas las posibilidades del juguete, se llegó a la conclusión de ser el robot ideal para las ideas que se plantearon en un principio.

3.4 Datos técnicos del Sphero 2.0

Aunque para este proyecto se ha utilizado la primera versión del Sphero, he aquí una descripción de los datos técnicos de la nueva entrega de Orbotix, el cual puede utilizarse en el futuro.

El nuevo motor de inteligencia artificial llamado Superdrive se incluye dentro la última versión del Sphero, publicada el 30 de agosto de 2013. Gracias a esta característica, la velocidad máxima del Sphero se eleva hasta los 2,13 metros por segundo (7,2 km/h).

El LED multicolor, capaz de proyectar más de 16 millones de colores, es tres veces más brillante. Como últimas novedades técnicas, también se ha modificado el centro de gravedad, y, se ha hecho más resistente a los golpes.

El fabricante Orbotix ha incluido dos rampas en la caja de las bolas Sphero 2.0, las cuales amplían el rango de posibilidades a la hora de mover la bola, además de poder utilizar diferentes carcasas para obtener mayor tracción en algunas superficies.

A nivel de software, la empresa ha puesto más de 25 aplicaciones y juegos de cara al usuario, ampliando claramente su oferta frente a la versión predecesora.



Figura 3.4: Imagen del Sphero 2.0 con sus nuevas carcasas.



Figura 3.5: Imagen del Sphero 2.0 con las nuevas rampas y el cargador

Capítulo 4: De Flurry a Adaptare

En este capítulo se exponen las razones por la cuales se ha dejado de utilizar el producto Flurry Analytics, pasando a utilizar un servicio específico llamado Adaptare para la recogida de datos de utilización.

4.1 Flurry Analytics

Flurry es una empresa fundada en el año 2005, que, entre sus diferentes actividades, proporciona una librería para uso en dispositivos Android e iOS. Mediante el uso de esta librería, la compañía permite la posibilidad de realizar análisis de las interacciones de los consumidores con las aplicaciones para teléfonos inteligentes o Tablets PC.

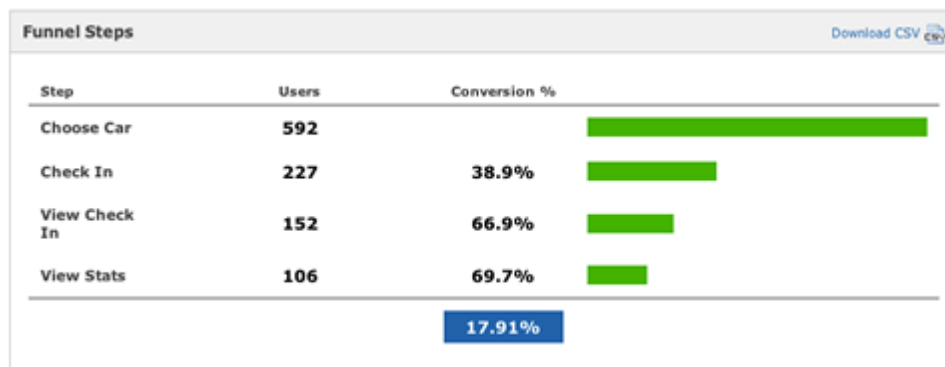


Figura 4.1: Representación de un Funnel Analysis

Para la primera versión de la aplicación de control del Sphero se utilizó este producto para recoger los datos de uso de la misma. Una vez utilizado en la práctica, los resultados son correctos, pero muy genéricos. De esta forma, ninguno de los datos que se pueden almacenar son de interés educativo, como los colores o mandos utilizados por el usuario, son almacenados. Teniendo como objetivo la mejora de esta parte, se ha decidido realizar un producto propio, mediante el cual se realice la misma tarea, pero con datos específicos. De esta forma nace Adaptare.

4.2 Adaptare

Adaptare es una aplicación web que trata de ser la solución requerida por las personas que quieran analizar los comportamientos del usuario con la aplicación móvil de control del dispositivo robótico Sphero. Para ello, se ha dividido el trabajo en tres frentes, tratando de cubrir todo lo necesitado para este objetivo:

- **Base de datos:** donde se recogen todos los movimientos interesantes de carácter educativo de la aplicación.
- **Inserción en la base de datos:** a utilizar dentro de la aplicación sobre Android, son llamados con los datos a guardar, encargándose de almacenarlos en la base de datos.
- **Aplicación web:** mediante la cual se pueden consultar las estadísticas de uso de la aplicación móvil sobre Android.

Con este conjunto, se tiene una imagen mucho más cercana a la utilización del Sphero, de forma que las personas encargadas de cuidar y educar a personas con discapacidad puedan tener análisis de los comportamientos realizados con la aplicación, y corregir los que no consideren adecuados. Por otra parte, estas estadísticas pueden abrir puertas a más campos de investigación, llegando a aplicar nuevas tecnologías para la corrección de esos comportamientos incorrectos indicados anteriormente.

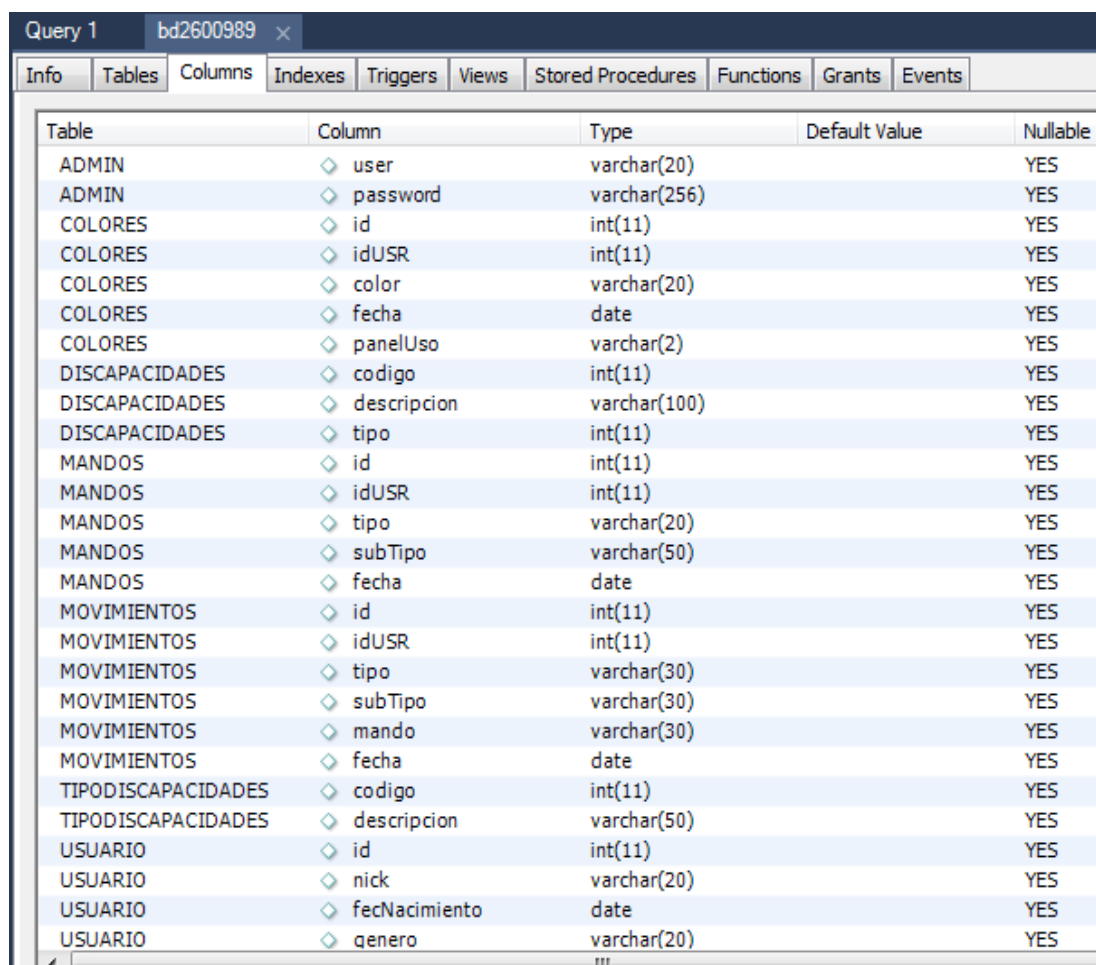


Figura 4.2: Sistema completo de Adaptare más la aplicación Android que la utiliza

4.3 Base de datos

La base de datos que almacena todas las estadísticas de uso del Sphero es una sencilla base de datos, que contiene siete tablas, con las que se puede tener un reflejo del uso de la aplicación.

Para la realización de esta base de datos se ha elegido el producto MySQL, a día de hoy, propiedad de Oracle Corporation. La principal razón de uso de MySQL es que se ofrece bajo licencia pública general de GNU, ya que, por las características de esta base de datos, no exige unos mínimos requerimientos técnicos. De lo contrario, el análisis hubiera sido más exhaustivo y tal vez la decisión hubiera sido otra.



| Table | Column | Type | Default Value | Nullable |
|--------------------|---------------|--------------|---------------|----------|
| ADMIN | user | varchar(20) | | YES |
| ADMIN | password | varchar(256) | | YES |
| COLORES | id | int(11) | | YES |
| COLORES | idUSR | int(11) | | YES |
| COLORES | color | varchar(20) | | YES |
| COLORES | fecha | date | | YES |
| COLORES | panelUso | varchar(2) | | YES |
| DISCAPACIDADES | codigo | int(11) | | YES |
| DISCAPACIDADES | descripcion | varchar(100) | | YES |
| DISCAPACIDADES | tipo | int(11) | | YES |
| MANDOS | id | int(11) | | YES |
| MANDOS | idUSR | int(11) | | YES |
| MANDOS | tipo | varchar(20) | | YES |
| MANDOS | subTipo | varchar(50) | | YES |
| MANDOS | fecha | date | | YES |
| MOVIMIENTOS | id | int(11) | | YES |
| MOVIMIENTOS | idUSR | int(11) | | YES |
| MOVIMIENTOS | tipo | varchar(30) | | YES |
| MOVIMIENTOS | subTipo | varchar(30) | | YES |
| MOVIMIENTOS | mando | varchar(30) | | YES |
| MOVIMIENTOS | fecha | date | | YES |
| TIPODISCAPACIDADES | codigo | int(11) | | YES |
| TIPODISCAPACIDADES | descripcion | varchar(50) | | YES |
| USUARIO | id | int(11) | | YES |
| USUARIO | nick | varchar(20) | | YES |
| USUARIO | fecNacimiento | date | | YES |
| USUARIO | genero | varchar(20) | | YES |

Figura 4.3: Tablas de la base datos vistas con MySQL Workbench 6.1

Se han creado estas siete tablas:

- **ADMIN:** contiene el usuario y la password para el acceso a Adaptare.

Campos:

- user VARCHAR(20): código de usuario de acceso a Adaptare.
- password VARCHAR(256): contiene la contraseña de acceso a Adaptare.

- **USUARIO:** listado de usuarios de la aplicación.

Campos:

- id int(11): identificador del elemento en la tabla.
- nick VARCHAR(20): identificador del usuario.
- fecNacimiento DATE: fecha de nacimiento del usuario.
- primeraDiscapacidad int(11): primera discapacidad del usuario.
- segundaDiscapacidad int(11): en el caso de tener más de una, está este segundo campo.
- terceraDiscapacidad int(11): completando el anterior, una tercera posibilidad.
- genero VARCHAR(20): masculino o femenino

- **TIPODISCAPACIDADES:** clasificación de discapacidades:

Campos:

- codigo int(11): código identificativo del tipo de discapacidad.
- descripcion VARCHAR(50): descripción del tipo de discapacidad.

- **DISCAPACIDADES:** desglose de los subtipos de discapacidades:

Campos:

- codigo int(11): código identificativo del subtipo de discapacidad.
- descripcion VARCHAR(100): descripción del subtipo de discapacidad.
- tipo int(11): código del tipo de discapacidad a la que pertenece.

- **MANDOS:** relación de mandos seleccionados por el usuario.

Campos:

- id int(11): identificador de la entrada en la tabla.
- idUSR int(11): identificador del usuario que está utilizando el mando.
- tipo VARCHAR(20): tipo de mando utilizado por el usuario.
- subTipo VARCHAR(50): completa la información del tipo de mando.
- fecha DATE: fecha en la que ha sido utilizado el mando.

- **COLORES:** relación de colores seleccionados por el usuario.

Campos:

- id int(11): identificador de la entrada en la tabla.
- idUSR int(11): identificador del usuario que está utilizando el mando de colores.
- color VARCHAR(20): color seleccionado por el usuario.
- fecha DATE: fecha en la que ha sido utilizado alguno de los mandos de colores.
- panelUso VARCHAR(2): PO, mando de Porciones, PA, mando de Paleta.

- **MOVIMIENTOS:** relación de movimientos realizados por el usuario.

Campos:

- id int(11): identificador de la entrada en la tabla.
- idUSR int(11): identificador del usuario que ha realizado el movimiento.
- tipo VARCHAR(20): tipo de movimiento realizado por el usuario.
- subTipo VARCHAR(50): completa la información del tipo de movimiento.
- fecha DATE: fecha en la que se ha realizado el movimiento.

Dos detalles importantes de las tablas de esta base de datos son los siguientes:

- **Identificadores:** para obtener un valor no repetido para cada identificador de cada una de las entradas en las tablas se ha utilizado el valor del Timestamp del momento.
- **Fechas:** las fechas almacenadas corresponden a la que en ese momento tiene el servidor de la base de datos, de forma que pueden diferir de la mostrada en el dispositivo Android que tiene instalada la aplicación de manejo de Sphero.

4.4 Inserción en la base de datos

La realización de estos servicios responde a la necesidad de ser llamados desde la aplicación sobre Android para almacenar las estadísticas de uso en la base de datos MySQL creada para tal efecto. Están realizados en el lenguaje de programación PHP, en concreto, en su versión 5, aunque el código escrito existe desde versiones anteriores.

Se ha decidido utilizar el lenguaje de programación PHP por formar parte del software libre publicado bajo licencia PHP. La licencia PHP es la licencia bajo la que se publica el lenguaje de programación PHP. De acuerdo a la Free Software Foundation es una licencia de software libre no copyleft y una licencia de código abierto según la Open Source Initiative. Debido a la restricción en el uso del término "PHP", no es compatible con la licencia pública general de GNU.

Por otra parte, la facilidad de uso, además el potencial del mismo, son otras de las razones para la utilización de PHP como lenguaje de programación de los servicios web necesarios para la aplicación sobre Android.

Su funcionamiento es sencillo: reciben los parámetros desde la aplicación sobre Android, establecen una conexión con la base de datos, ejecutan la sentencia SQL para almacenar los datos, y cierran la conexión con la base de datos.

```
//Conexión con la BBDD
$link = mysql_connect('mysql.adaptare.es', 'ADM_bd2600989', 'Upm-2014');
mysql_select_db("bd2600989", $link);

//Control de errores
if (mysql_errno()) {
    echo "Failed to connect to MySQL: " . mysql_connect_error();
}
```

Figura 4.4: Ejemplo de conexión con la base de datos desde un servicio PHP

```
//Ejecuta la query
mysql_query($insert);
//Cierre
mysql_close($link);

?>
```

Figura 4.5: Ejemplo de ejecución de query y desconexión con la base de datos desde un servicio PHP

He aquí la relación de los mismos, además del código básico de uno de ellos, el cual es repetido en los otros casos, cambiando los campos necesarios en cada caso:

- **setColores:** realiza la conexión con la base de datos y ejecuta la sentencia de inserción de los colores que están siendo utilizados en los mandos que disponen de los mismos en la aplicación sobre Android.
- **setMandos:** realiza la conexión con la base de datos y ejecuta la sentencia de inserción de los mandos que están siendo utilizados por el usuario en la aplicación que controla el Sphero.
- **setMovimientos:** realiza la conexión con la base de datos y ejecuta la sentencia de inserción de los movimientos que está realizando el Sphero, previamente ordenados desde la aplicación sobre Android.
- **setUsuario:** inserta o actualiza los datos del usuario que está utilizando el Sphero. Esta función se encuentra en el menú del Profesor, y será descrita más adelante.

Los parámetros a introducir en la base de datos son recibidos como parámetros en la URL de llamada a los mismos:

```
<?php
//Parámetros en URL
$nick = $_GET['nick'];
$fecNacimiento = $_GET['fecNacimiento'];
$fecNacimiento2 = substr($fecNacimiento, -4, 4)."-".substr($fecNacimiento, -7, 2)."-".substr($fecNacimiento, -10, 2);
$genero = $_GET['genero'];
$primeraDiscapacidad = $_GET['primeraDiscapacidad'];
$segundaDiscapacidad = $_GET['segundaDiscapacidad'];
$terceraDiscapacidad = $_GET['terceraDiscapacidad'];
```

Figura 4.5: Ejemplo en la recepción de parámetros en el servicio setUsuario

4.5 Aplicación Web

La aplicación web realizada tiene como funcionalidad básica la consulta de los datos registrados en la base de datos MySQL anteriormente descrita. Programada en PHP y HTML5, su diseño es muy básico, pero responde a las necesidades de consultar los datos interesantes para el papel educativo del proyecto.

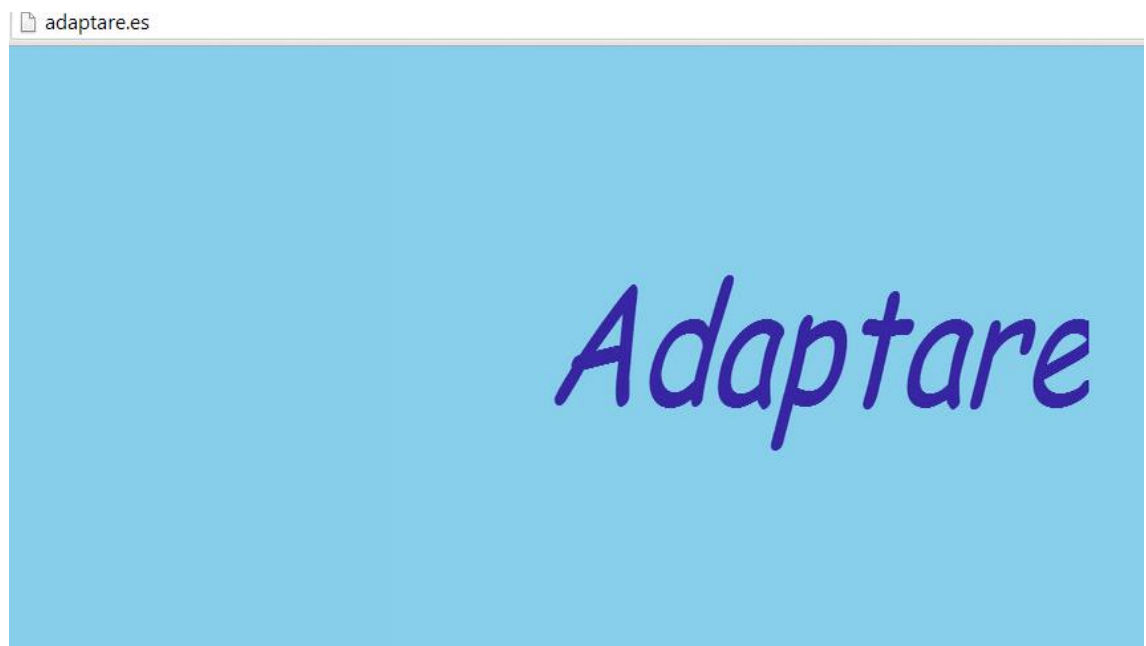


Figura 4.6: Logotipo de Adaptare

La aplicación se divide en cinco partes:

- Acceso: el usuario debe introducir un usuario y una contraseña.
- Menú: el usuario debe elegir el tipo de consultas que desea realizar.
- Listado de usuarios: relación de los usuarios registrados para el uso de la aplicación.
- Ventana de búsquedas: en función de si son colores, mandos o movimientos los parámetros de búsqueda son diferentes. Por ello, se han realizado tres ventanas diferentes.

- Ventana de resultados: en función de si son colores, mandos o movimientos, los parámetros de las tablas de resultados son diferentes. Por ello, se han realizado tres ventanas diferentes.

Completando las ventanas de resultados, estos pueden exportarse a un documento CVS y así poder consultarlo y analizarlos en otros medios.

Esta es la descripción en profundidad de cada una de las ventanas:

- **Acceso**



Figura 4.6: Acceso a Adaptare

Una vez pulsado el logotipo de Adaptare de la ventana de acceso, la pantalla de acceso es muy sencilla: dos campos de texto para introducir un usuario y la contraseña correspondiente y un botón Aceptar, mediante el cual se envían los datos a validar. En el caso de ser correctos, se accederá al menú, y, de lo contrario, no se podrá acceder a la aplicación de consulta de estadísticas.

Para garantizar la seguridad se ha utilizado el algoritmo de hash seguro SHA-1. Publicada en el año 1995, codifica una cadena de caracteres de forma que su lectura resulta complicada en caso de recibir un ataque en el sistema de almacenamiento de los datos.

- **Menú**



Figura 4.7: Menú de Adaptare

Dentro de la venta del menú, aparecen cuatro enlaces. El aspecto visual es sencillo, debido a que se le ha dado importancia a cumplir los requisitos funcionales, ya que en este caso su aporte es fundamental.

Tal y como aparece en la imagen, este menú lo forman cuatro enlaces: Usuarios, que abre una ventana con el listado de usuarios registrados en la aplicación, Colores, que abre una ventana para filtrar las estadísticas de los colores utilizados por el usuario en la aplicación de control del Sphero, Movimientos, que abre una ventana para filtrar las estadísticas de los movimientos realizados por el usuario en la aplicación, y, Mandos, el cual también abre una ventana para la consulta de la relación de mandos utilizados en la aplicación del juguete teledirigido Sphero.

- **Listado de usuarios**

En el menú anterior se puede elegir la opción de usuarios. En este caso, se mostrará un listado de todos los usuarios registrados que están utilizando o han utilizado la aplicación sobre Android para el manejo del Sphero. Al final del listado, existe el botón Exportar a CVS, que, al ser pulsado, generará un fichero CVS con los resultados obtenidos en la búsqueda de los usuarios.

El listado de usuarios muestra los siguientes datos de cada uno de ellos: nick, fecha de nacimiento, género, primera discapacidad, segunda discapacidad y tercera discapacidad:

| Nick | Fecha Nacimiento | Género | Primera Discapacidad | Segunda Discapacidad | Tercera Discapacidad |
|-------|------------------|-----------|----------------------|----------------------|----------------------|
| Lera3 | 26-07-1982 | Masculino | Demencias | Sin discapacidad | Sin discapacidad |
| Lera | 26-07-1982 | Masculino | Mala visión | Sin discapacidad | Sin discapacidad |
| Lera1 | 26-07-1982 | Masculino | Mala visión | Sin discapacidad | Sin discapacidad |

Volver

Exportar a CSV

Figura 4.8: Listado de usuarios de la aplicación Android

- **Búsquedas**

Las búsquedas de los datos de utilización de la aplicación se han dividido en tres bloques, como puede observarse en el menú de la aplicación: colores, mandos y movimientos. Los tres tienen una serie de parámetros en común, como el Nick de usuario o las fechas de utilización. En el caso de la búsqueda de colores, se añade el parámetro color y el mando utilizado: paleta de colores o porciones. En la consulta de movimientos, los parámetros diferentes son el tipo de movimiento, el subtipo del mismo y el mando utilizado. Por último, en la búsqueda de mandos también se añaden el tipo y subtipo de los mismos. Para todos los casos, el único parámetro obligatorio en las búsquedas es el ID Usuario:

ID Usuario:

Tipo:

Subtipo:

Fecha inicial:

dd/mm/aaaa

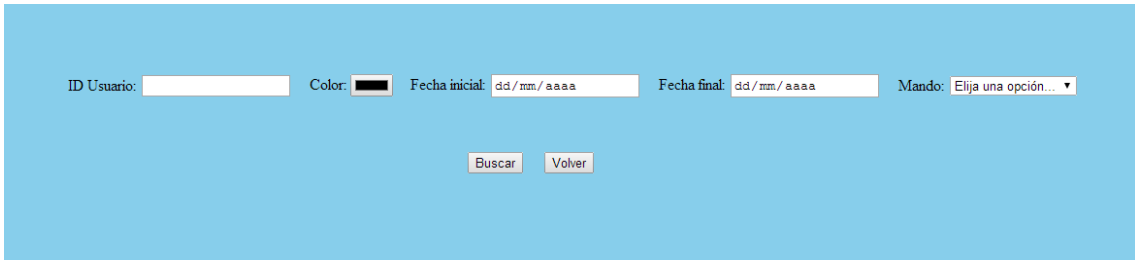
Fecha final:

dd/mm/aaaa

Buscar

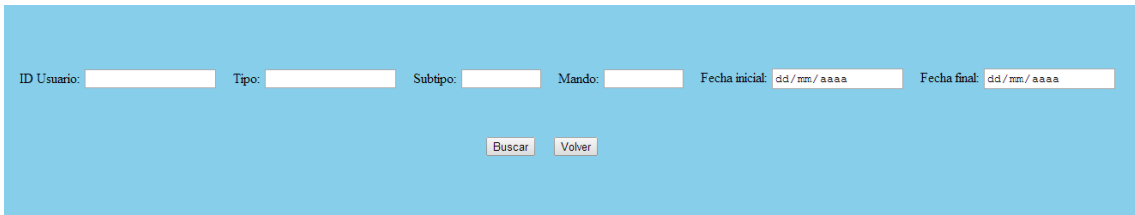
Volver

Figura 4.9: Búsqueda de mandos utilizados



Formulario de búsqueda de colores utilizados. El formulario está sobre un fondo azul claro y contiene los siguientes campos: 'ID Usuario:' con un campo de texto blanco; 'Color:' con un selector de color que muestra un cuadrado negro; 'Fecha inicial:' con un campo de texto blanco que muestra el formato 'dd/mm/aaaa'; 'Fecha final:' con un campo de texto blanco que muestra el formato 'dd/mm/aaaa'; y 'Mando:' con un menú desplegable que muestra 'Elija una opción...'. Debajo de estos campos hay dos botones: 'Buscar' y 'Volver'.

Figura 4.10: Búsqueda de colores utilizados



Formulario de búsqueda de movimientos realizados. El formulario está sobre un fondo azul claro y contiene los siguientes campos: 'ID Usuario:' con un campo de texto blanco; 'Tipo:' con un campo de texto blanco; 'Subtipo:' con un campo de texto blanco; 'Mando:' con un campo de texto blanco; 'Fecha inicial:' con un campo de texto blanco que muestra el formato 'dd/mm/aaaa'; y 'Fecha final:' con un campo de texto blanco que muestra el formato 'dd/mm/aaaa'. Debajo de estos campos hay dos botones: 'Buscar' y 'Volver'.

Figura 4.11: Búsqueda de movimientos realizados

Los parámetros seleccionados se enviarán a través de la URL a la ventana de resultados, donde se realiza la conexión con la base de datos y la respuesta con los mismos.

- **Resultados**

De forma similar a las búsquedas, también se han diferenciado tres pantallas de resultados, en función de si se están buscando colores, movimientos o mandos. Los resultados se muestran en la pantalla, mediante tablas en HTML, con la posibilidad de exportarlos a un documento con formato CVS.

Para la obtención de resultados de uso, el funcionamiento es similar al indicado anteriormente en el listado de usuarios: recepción de parámetros, creación de la conexión con la base de datos, creación de la query de inserción de datos y cierre de la conexión con la base de datos.
























| | | | |
|------|---|------------|-----------|
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Porciones |
| Lera |  | 23-08-2014 | Porciones |
| Lera |  | 23-08-2014 | Porciones |
| Lera |  | 23-08-2014 | Porciones |
| Lera |  | 23-08-2014 | Porciones |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |
| Lera |  | 23-08-2014 | Paleta |

Figura 4.12: Resultado de la búsqueda de colores utilizados en la aplicación Android.

| | | | | |
|------|----------|---------------|--------------|------------|
| Lera | Adelante | Correccion:30 | Posicion | 23-08-2014 |
| Lera | Derecha | Correccion:30 | Posicion | 23-08-2014 |
| Lera | Noroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Noreste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Sureste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Suroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Noroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Noreste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Noroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Suroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Sureste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Suroeste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Sureste | Dibujada | Trayectoria | 23-08-2014 |
| Lera | Sur | | Acelerometro | 23-08-2014 |
| Lera | Norte | | Acelerometro | 23-08-2014 |
| Lera | Oeste | | Acelerometro | 23-08-2014 |
| Lera | Este | | Acelerometro | 23-08-2014 |
| Lera | Sur | | Acelerometro | 23-08-2014 |
| Lera | Norte | | Acelerometro | 23-08-2014 |
| Lera | Oeste | | Acelerometro | 23-08-2014 |
| Lera | Este | | Acelerometro | 23-08-2014 |
| Lera | Sur | | Acelerometro | 23-08-2014 |
| Lera | Norte | | Acelerometro | 23-08-2014 |
| Lera | vibra | Fijo | Voz | 23-08-2014 |
| Lera | Sur | | Acelerometro | 23-08-2014 |

Figura 4.13: Resultado de la búsqueda de movimientos realizados.

| | | | |
|------|--------------|-------|------------|
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Posicion | | 23-08-2014 |
| Lera | Posicion | | 23-08-2014 |
| Lera | Locator | | 23-08-2014 |
| Lera | Posicion | | 23-08-2014 |
| Lera | Posicion | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Posicion | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Paleta | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Porciones | | 23-08-2014 |
| Lera | Trayectoria | | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Voz | Macro | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |
| Lera | Paleta | | 23-08-2014 |
| Lera | Acelerometro | | 23-08-2014 |

Volver Exportar a CSV

Figura 4.14: Resultado de la búsqueda de mandos utilizados.

Capítulo 5: Discapacidad

A lo largo de este capítulo dedicado a la discapacidad, se ahondará en la comunicación, con los trastornos que impiden una comunicación óptima, para completarlo con los tipos de discapacidad, detallando cada uno de ellos. Para finalizar, se incluirán sistemas de comunicación, juguetes adaptados y la accesibilidad en dispositivos móviles, ya que todo ello forma parte de la aplicación.

5.1 La comunicación

5.1.1 Definición

Un primer acercamiento a la definición de comunicación puede realizarse desde su etimología. La palabra deriva del latín *communicare*, que significa “compartir algo, poner en común”. Por lo tanto, la comunicación es un fenómeno inherente a la relación que los seres vivos mantienen cuando se encuentran en grupo.

Cuando los seres humanos se comunican entre sí están compartiendo cuestiones, llevando a que las situaciones propias sean comunes con el otro y con las que este tenga. Por tanto, la comunicación es una actividad absolutamente humana y parte de la relación de las personas en cualquier ámbito y momento de la vida.

Si no fuese gracias a la comunicación no podría conocerse lo que nos rodea y además compartirlo con el propio entorno, pero al ser un hecho concreto y a disposición de todos, la comunicación facilita la obtención de información para conocer, expresarse y relacionarse con el resto de las personas.

El proceso de comunicación implica la emisión de señales, tales como sonidos, gestos o señas, con la única intención de dar a conocer un mensaje. Para que en la comunicación, el mensaje, al destinatario, será necesario que este cuente con las habilidades de decodificar e interpretar el mensaje en cuestión.

En tanto, en este proceso casi siempre se producirá un *feedback*, una ida y vuelta, porque una vez que el emisor emite su mensaje el proceso se revierte y el receptor al momento de responder se convertirá en el emisor, siendo el emisor original el receptor del proceso de comunicación.

Sus elementos, por tanto son:

- **Emisor:** Aquél que transmite la información (un individuo, un grupo o un dispositivo).
- **Receptor:** Aquél, individual o colectivamente, que recibe la información. Puede ser un dispositivo.
- **Código:** Conjunto o sistema de signos que el emisor utiliza para codificar el mensaje.
- **Canal:** Elemento físico por donde el emisor transmite la información y que el receptor capta por los sentidos corporales. Se denomina canal tanto al medio natural (aire, luz) como al medio técnico empleado (impresión, telegrafía, radio, teléfono, televisión, ordenador, etc.) y se perciben a través de los sentidos del receptor (oído, vista, tacto, olfato y gusto).
- **Mensaje:** La propia información que el emisor transmite.
- **Contexto:** Circunstancias temporales, espaciales y socioculturales que rodean el hecho o acto comunicativo y que permiten comprender el mensaje en su justa medida.

La principal dificultad que puede presentarse dentro del proceso comunicativo es lo que se conoce como ruido, una perturbación que complicará el normal desarrollo del mensaje. Algunos ruidos comunes a la hora de comunicarse resultan ser: distorsión en el sonido, empleo de ortografía defectuosa o la disfonía del emisor.

La comunicación, en los seres humanos resulta ser una acción propia de la actividad psíquica, procediendo la misma del pensamiento, el lenguaje y las capacidades psicosociales de relación. La comunicación, ya sea verbal o no, le permitirá a los individuos influir en las decisiones de los demás y también ser influido por las que tenga el resto.

5.1.2 Factores que impiden la comunicación humana

El conjunto de factores que impiden la comunicación son considerados como barreras u obstáculos, que afectan sus objetivos, llegando a deformarlos y entre ellas podemos mencionar: *“Barreras psicológicas, semánticas, fisiológicas, físicas, administrativas”*.

- **Psicológicas.** Agrupan a varios procesos o factores mentales que no permiten la comprensión de una idea, como por ejemplo; las suposiciones, preocupaciones o emociones externas al ámbito laboral, timidez, sobre valoración, etc. Estos procesos afectan a un receptor para que pueda aceptar o rechazar la comunicación de ideas, ya que para convencer, se debe explicar y las causas que pueden motivar a la formación de dichas barreras u obstáculos pueden ser; el estatus de poder, el sarcasmo, las críticas, el conocimiento exacto de un tema, habilidad lingüística, aspecto físico, interrupciones continuas, etc.
- **Semánticas o verbales.** Definidas por no saber precisar un sentido, a través del significado de las palabras, generando distintas posibles interpretaciones entre emisor y receptor.
- **Fisiológicas.** Cuando afectan los sentidos del emisor o el receptor, impidiéndole la comprensión, como por ejemplo deficiencias foniátricas.
- **Físicas.** se refieren a los factores ambientales, como el ruido, la distancia, visual, telecomunicación en sí.
- **Administrativas.** Son las que afectan a las estructuras organizacionales, por ser deficientes a nivel operacional de la empresa, así como inadecuadas o faltas de estrategia.

5.1.3 Trastornos en la comunicación y el lenguaje

Los trastornos que se pueden presentar en este ámbito, son muchos y variados, pueden afectar a uno, a varios o a todos los componentes del lenguaje, difieren en su etiología, en el pronóstico, en las necesidades educativas que generan y en la respuesta educativa que requieren.

Delimitar el concepto de trastorno, dependerá de dónde se ponga el límite de lo normal y lo patológico. Por lo tanto al ser subjetivo, dependerá del criterio del observador que va a emitir el juicio y de las normas sociales imperantes.

En general, se considera que un lenguaje normal, es aquel que tiene un uso preciso de las palabras según su significado, un vocabulario de calidad y cantidad, claridad de la articulación, una forma gramatical adecuada, un ritmo y velocidad apropiados, un volumen de voz audible, un tono adecuado a la edad y el sexo y una entonación de las frases en concordancia con su significado y sus necesidades expresivas.

Este canon de normalidad, sólo es aplicable al lenguaje adulto, ya que en el lenguaje infantil normal, todas o casi todas estas habilidades están en pleno proceso de desarrollo, sin que se considere un trastorno, sino propio del desarrollo evolutivo y que de forma natural o con intervención directa o indirecta, irá desapareciendo sin dejar secuelas.

No obstante, existe un pequeño grupo de niños y niñas que sí presentan verdaderos indicadores de trastornos. De aquí se deduce la importancia de conocer los parámetros evolutivos de la edad, para no incurrir en errores de considerar patológico lo que es normal en determinados momentos del proceso evolutivo.

5.1.3.1 Trastornos en el habla

Hablar es expresar a través de mecanismos físicos y fisiológicos todos los procesos de lenguaje interior (léxico-semánticos, morfológico-sintáctico, fonológico y pragmático). Para hablar, se necesita un flujo de aire en la espiración que al pasar por las cuerdas vocales, las hace vibrar y producir el sonido, las posiciones y movimientos de los órganos de la articulación (labios, mandíbula, lengua, paladar, etc.), así como, la forma en que se emite el aire (oral o nasal), y, cómo resuena en las cavidades orales y nasales, es lo que permite producir el habla.

Clasificación:

- Alteraciones que afectan a la articulación
 - Dislalias
 - Trastorno Fonológico

- Disglosias
- Disartrias
- Alteraciones que afectan a la fluidez verbal y el ritmo en la expresión
 - Disfemia
 - Taquilalia y farfullero
 - Bradilalia
- Alteraciones de la voz
 - Disfonía
 - Rinofonía

5.1.3.2 Trastornos del lenguaje

Los trastornos del lenguaje, se dan formando un continuo que iría, desde el retraso simple del lenguaje hasta la pérdida total de las capacidades lingüísticas en el caso de la afasia.

Otro importante aspecto a resaltar, es el diferente pronóstico que cada uno de los tipos de trastorno tiene. Mientras, en los Retrasos en la adquisición del lenguaje, el pronóstico es bueno, pudiendo normalizarse con la intervención, Rondal, (1985) en los casos de Trastorno Específico del Lenguaje o en Afasia, el pronóstico es negativo, ya que aunque mejoran, suelen dejar secuelas permanentes.

Clasificación:

- Retrasos en la adquisición y desarrollo del lenguaje
- Trastorno Específico del Lenguaje
- Afasias

5.1.3.3 Trastornos de la comunicación

A lo largo del siglo pasado, se puso el énfasis en el estudio de las alteraciones que presentaban los niños en su desarrollo del lenguaje. Inicialmente, se centraron en aquellos que perturbaban la voz y el habla. En la siguiente fase con el auge de la lingüística, el enfoque se centró en el estudio de estructuras semánticas, gramática y sintaxis. Al final del siglo, con el enfoque pragmático, se pretendió situar su desarrollo en el marco de comunicación y de las interacciones sociales.

Ante esta denominación, se estaría frente a trastornos del componente pragmático del lenguaje. Es un concepto reciente, en el que todavía no existe un claro consenso en la clasificación de los trastornos o síndromes que lo integran.

5.2 Discapacidades

5.2.1 Discapacidades psíquicas

El término “discapacidad intelectual” equivale a “retraso mental” que se define por la AAMR a la “discapacidad caracterizada por limitaciones significativas en el funcionamiento intelectual y en la conducta adaptativa que se manifiesta en habilidades conceptuales, sociales y prácticas. Esta discapacidad comienza antes de los 18 años.”

El desarrollo intelectual estará relacionado, según este organismo, con las siguientes dimensiones:

- Capacidades intelectuales
- Conducta adaptativa (conceptual, social y práctica)
- Participación, interacciones y roles sociales
- Salud (salud física, salud mental, etiología)
- Contexto (ambientes y cultura).

Se puede realizar la siguiente clasificación atendiendo al índice del C.I., según una escuela Navarra (CREENA) especializada en gente con discapacidad psíquica:

- **Retraso mental/discapacidad intelectual ligera o leve.** Se incluyen en la misma las personas cuya medida en C.I., sin llegar a 55 – 50, se sitúa por debajo de 75 – 70 (unas 2 desviaciones típicas por debajo de la media, con un error de medida de aproximadamente 5 puntos). Acerca de ese tramo límite por arriba, se indica que se podría diagnosticar discapacidad ligera con un cociente intelectual entre 70 y 75 si existe déficit significativo en conducta adaptativa, pero no cuando no exista.

Estas personas suelen presentar ligeros déficits sensoriales y/o motores, adquieren habilidades sociales y comunicativas en la etapa de educación infantil y, con frecuencia, no se diferencian de sus iguales por los rasgos físicos. A lo largo de la enseñanza básica, suelen llegar a adquirir aprendizajes instrumentales y algún grado de conocimientos académicos.

- **Retraso mental/discapacidad intelectual media o moderada.** Se incluyen en la misma las personas cuya medida en C.I. está entre 55 – 50 y 40 – 35. Lógicamente, con este nivel intelectual, las posibilidades adaptativas de estas personas suelen verse muy afectadas en todas las áreas de desarrollo. Como grupo suponen alrededor del 10 % de toda la población con discapacidad intelectual.

A lo largo de la enseñanza básica, suelen desarrollar habilidades comunicativas durante los primeros años de la infancia y, durante la escolarización, pueden llegar a alcanzar algún grado de aprendizajes instrumentales. Suelen aprender a trasladarse de forma autónoma por lugares que les resulten familiares, atender a su cuidado personal con cierta supervisión y beneficiarse del adiestramiento en habilidades sociales.

- **Retraso mental / discapacidad intelectual severa o grave.** Se sitúa en el intervalo de C.I. entre 35 – 40 y 20 – 25 y supone el 3 – 4 % del total de la discapacidad. Las adquisiciones de lenguaje en los primeros años suelen ser escasas y a lo largo de la enseñanza básica pueden aprender a hablar o a emplear algún signo de comunicación alternativo. Las

posibilidades adaptativas están muy afectadas en todas las áreas de desarrollo, pero es posible el aprendizaje de habilidades elementales de cuidado personal.

- **Retraso mental discapacidad profunda / pluridiscapacidad.** La mayoría de estas personas presenta una alteración neurológica identificada que explica esta discapacidad, la confluencia con otras (de ahí el término pluridiscapacidad que aquí se le asocia) y la gran diversidad que se da dentro del grupo. Ello condiciona el hecho de que uno de los ámbitos de atención prioritaria sea el de la salud física.

El C.I. de estas personas queda por debajo de 20 – 25 y son el 1 – 2 % de la tipología. Suelen presentar limitado nivel de conciencia y desarrollo emocional, nula o escasa intencionalidad comunicativa, ausencia de habla y graves dificultades motrices. El nivel de autonomía, si existe, es muy reducido.

5.2.2 Discapacidades motoras

Una persona con discapacidad motora es aquella que presenta de manera transitoria o permanente alguna alteración de su aparato motor, debido a un deficiente funcionamiento en el sistema nervioso, muscular y /u óseo, o en varios de ellos relacionados, que en grados variables limita alguna de las actividades que puede realizar el resto de las personas de su edad.

Esta discapacidad implica la limitación del normal desplazamiento físico. Las personas que tienen este tipo de discapacidades pueden ser semiambulatorias o no ambulatorias. En el caso del primer tipo, se movilizan ayudadas por elementos complementarios, como ser muletas, bastones, andadores, etc. Las no ambulatorias sólo pueden desplazarse con silla de ruedas. Esto implica la fundamental importancia de estos elementos para las personas con discapacidad.

Las causas de esta discapacidad pueden ser por secuelas neurológicas, miopáticas, ortopédicas o reumatológicas. Las secuelas neurológicas se dividen en cerebrales (parálisis o hemiplejia) o medulares.

Para clasificar una deficiencia motora hay que atender a cuatro criterios:

Fecha de aparición

La deficiencia puede aparecer desde el nacimiento, después, en la adolescencia y a lo largo de toda la vida:

- Congénitas: cualquier tipo de malformaciones. Por ejemplo: espina bífida.
- Después del nacimiento: como la parálisis cerebral, cuya detección suele ser unas semanas más tarde del nacimiento (cuando no muestra los patrones motrices adecuados a su edad).
- Adolescencia: miopatías.
- A lo largo de toda la vida: accidentes, traumatismos (traumatismos craneoencefálicos, vertebrales, derrames, etc.) etc.

Etiopatología

- Transmisión genética: Miopatía de Duchenne de Boulogne, transmitida por una madre portadora
- Infecciones microbianas: tuberculosis ósea, poliomielitis, etc.
- Accidentes: pueden sobrevenir en el momento del parto o a lo largo de toda la vida
- Origen desconocido: como, por ejemplo: espina bífida, escoliosis ideopática o el origen de los tumores

Localización topográfica

- Parálisis: monoplejía, diplejía o parálisis bilateral, tetraplejía, paraplejía, triplejía y hemiplejía
- Paresia: es una parálisis más suave, ligera o incompleta. Tipos: monoparesia, hemiparesia, tetraparesia, paraparesia o paraparesia, triparesia y diparesia.

Origen

- Origen cerebral: parálisis cerebral, traumatismos cerebrales, tumores cerebrales, etc.

- Origen espinal: poliomielitis, espina bífida, lesiones medulares degenerativa que degeneran en ataxias y traumatismo medulares
- Origen Muscular: miopatías (en realidad son distrofias)
- Origen óseo-articular u osteoarticular: malformaciones congénitas, distrofias, enfermedades microbianas (osteomielitis o tuberculosis osteoarticular), traumatismos de la infancia, lesiones osteoarticulares por desviaciones del raquis (cifosis, escoliosis, etc.)

5.2.3 Discapacidades sensoriales

Una persona tiene un déficit sensorial cuando presenta una alteración o deficiencia que afecta a sus órganos sensoriales principales.

Dentro de la categoría de la discapacidad sensorial, se encuentra la discapacidad visual, la discapacidad auditiva y otros tipos de discapacidades relacionadas con disminución de algunos de los sentidos.

5.2.2.1 Discapacidad Visual

Según la Organización Mundial de la Salud una persona con deficiencia visual presenta una ausencia o mal funcionamiento del sistema óptico, causado por enfermedad, lesión o anomalía congénita que, a pesar de la corrección, convierte a la persona en un sujeto oficialmente considerado como deficiente visual en el país en el que vive.

La expresión más grave de una patología ocular es la ceguera, que se puede definir como una pérdida de visión lo suficientemente grande como para evitar que una persona se mantenga por sí misma en cualquier ocupación, volviéndolo dependiente de otros medios o personas para poder subsistir.

En términos generales, la mayoría de la gente asocia el significado de ciego con una persona con ausencia total de visión, sin embargo entre los conceptos de ceguera total y la visión normal existen distintas categorías. Según la OMS, la clasificación de la agudeza visual y deterioro es la siguiente:

- **La agudeza visual baja.** Significa visión entre 20/70 y 20/400 con la mejor corrección posible, o una visual de campo de 20 grados o menos.
- **Ceguera.** Se define como una agudeza visual peor de 20/400, con la mejor corrección posible, o un campo visual de 10 grados o menos
- **Moderado deterioro visual o baja visión.** Se considera agudeza visual de 20/70 a 20/400 (inclusive).

5.2.2.1 Discapacidad Auditiva

Disfunciones o alteraciones cuantitativas en una correcta percepción auditiva además trae aparejadas otras alteraciones cuya gravedad vendrá condicionada por factores tan importantes como son la intensidad de la pérdida auditiva y el momento de aparición de la misma.

Teniendo en cuenta que los órganos sensoriales proporcionan informaciones importantes que inciden en un desarrollo evolutivo adecuado de la persona, hay que considerar que el aislamiento y la falta de información a que se ve sometida ésta por causa del déficit auditivo pueden representar implicaciones importantes para su desarrollo del lenguaje y las diversas modalidades comunicativas, así como en los campos cognitivo, cognoscitivo, emocional, comportamental, social y ocupacional. Ocurren cuando hay un problema en los oídos o en una o más partes que facilitan la audición.

Una persona con una deficiencia auditiva puede ser capaz de oír algunos sonidos o puede no oír nada en absoluto. La palabra deficiencia significa que algo no está funcionando correctamente o tan bien como debería. La gente también utiliza palabras como sordo, sordera o duro de oído para referirse a las pérdidas auditivas. Es una de las anomalías congénitas más frecuentes. Los problemas auditivos también se pueden desarrollar más tarde en la vida de una persona.

Una de las clasificaciones que se pueden hacer acerca de los tipos de deficiencia auditiva son:

- Hipoacusias:
 - Deficiencia auditiva ligera.
 - Pérdida auditiva de entre 20 y 40 db.
 - Pequeñas dificultades articulatorias.
 - No identifican totalmente todos los fonemas.
 - Deficiencia auditiva media.
 - Pérdida auditiva de entre 40 y 70 db.
 - Identificación sólo de vocales.
 - Articulación defectuosa.
 - Lenguaje productivo limitado.
 - Capacidad para la estructuración del pensamiento verbal.
- Sorderas:
 - Deficiencia auditiva severa.
 - Pérdida auditiva de entre 70 y 90 db.
 - Percepción de algunos sonidos, pero imposibilidad de adquisición espontánea del lenguaje.
 - Los afectados son llamados sordos medios.
 - Deficiencia auditiva profunda.
 - Pérdida auditiva superior a los 90 db.
 - No pueden adquirir el lenguaje oral.
 - Los afectados son llamados sordos profundos.

5.3 Sistemas aumentativos y alternativos de comunicación (SAAC)

Los Sistemas Aumentativos y Alternativos de Comunicación (SAAC) son formas de expresión distintas al lenguaje hablado, que tienen como objetivo aumentar y/o compensar las dificultades de comunicación y lenguaje de muchas personas con discapacidad.

La comunicación y el lenguaje son esenciales para todo ser humano, para relacionarse con los demás, para aprender, para disfrutar y para participar en la sociedad y hoy en día, gracias a estos sistemas, no deben verse frenados a causa de las dificultades en el lenguaje oral. Por esta razón, todas las personas, ya sean niños, jóvenes, adultos o ancianos, que por cualquier causa no han adquirido o han perdido un nivel de habla suficiente para comunicarse de forma satisfactoria, necesitan usar un SAAC.

Entre las causas que pueden hacer necesario el uso de un SAAC, se encuentran la parálisis cerebral, la discapacidad intelectual, los trastornos del espectro autista, las enfermedades neurológicas tales como la esclerosis lateral amiotrófica, la esclerosis múltiple, el párkinson, las distrofias musculares, los traumatismos cráneo-encefálicos, las afasias o las pluridiscapacidades de tipologías diversas, entre muchas otras.

La Comunicación Aumentativa y Alternativa, no es incompatible, sino complementaria a la rehabilitación del habla natural, y además puede ayudar al éxito de la misma cuando éste es posible. No debe pues dudarse en introducirla a edades tempranas, tan pronto como se observan dificultades en el desarrollo del lenguaje oral, o poco después de que cualquier accidente o enfermedad haya provocado su deterioro. No existe ninguna evidencia de que el uso de CAA inhiba o interfiera en el desarrollo o la recuperación del habla.

5.4 Recursos utilizados en CAA

La Comunicación Aumentativa y Alternativa incluye:

- **Sistemas de símbolos.** Incluyen sistemas tanto gráficos (fotografías, dibujos, pictogramas, palabras o letras) como gestuales (mímica, gestos o signos manuales) y, en el caso de los primeros, requiere también el uso de productos de apoyo. Los diversos sistemas de símbolos se adaptan a las necesidades de personas con edades y habilidades motrices, cognitivas y lingüísticas muy dispares.
- **Productos de apoyo para la comunicación.** Incluyen recursos tecnológicos, como los comunicadores de habla artificial o los ordenadores personales y tablets con programas especiales, que permiten diferentes formas de acceso adaptadas algunas para personas con movilidad muy reducida, y facilitan también la incorporación de los diferentes sistemas de signos pictográficos y ortográficos, así como diferentes formas de salida incluyendo la salida de voz. También pueden consistir en recursos no tecnológicos, como los tableros y los libros de comunicación.
- **Estrategias y productos de apoyo para el acceso.** Son diversos instrumentos tales como los punteros, los teclados y ratones adaptados o virtuales y los conmutadores, que sirven para acceder a los ordenadores, comunicadores, tableros o libros de comunicación.

5.4.1 Sistemas de símbolos

Teniendo en cuenta una clasificación de los sistemas de símbolos para la CAA en gestuales y gráficos, se pueden encontrar una gradación desde sistemas muy sencillos, que se adaptan a personas con déficits cognitivos y lingüísticos de diversa consideración, hasta sistemas complejos que permiten niveles avanzados de lenguaje signado (basado en signos manuales) o asistido (basado en signos gráficos).

- **Símbolos gestuales.** Esta gradación abarca desde el uso de mímica y gestos de uso común hasta el uso de signos manuales, generalmente en el orden correspondiente al

lenguaje hablado; es lo que se denomina lenguaje signado o bimodal. Las lenguas de signos utilizadas por las personas no oyentes no se consideran SAAC, ya que constituyen idiomas que se han desarrollado y se adquieren de forma natural, al igual que ocurre con el lenguaje hablado. El uso de signos manuales requiere disponer de habilidades motrices suficientes, como puede ser el caso de personas con discapacidad intelectual o TEA.

- **Símbolos gráficos.** Abarcan desde sistemas muy sencillos basados en dibujos o fotografías hasta sistemas progresivamente más complejos como los sistemas pictográficos o la ortografía tradicional (letras, palabras y frases). Gracias a los productos de apoyo para la comunicación y los diversos recursos para el acceso, los sistemas gráficos pueden ser usados por personas con movilidad reducida, incluso en casos de extrema gravedad. Por ello, además de ser usados, como en el caso anterior, por personas con discapacidad intelectual o TEA, los usan también personas con discapacidades motoras.
- **Sistemas pictográficos.** Se aplican a personas que no están alfabetizadas a causa de la edad o la discapacidad. Tienen la ventaja de permitir desde un nivel de comunicación muy básico, que se adapta a personas con niveles cognitivos bajos o en etapas muy iniciales, hasta un nivel de comunicación muy rico y avanzado, aunque nunca tan completo y flexible como el que se puede alcanzar con el uso de la lengua escrita. Los sistemas pictográficos más usados en los diversos territorios del estado español son el sistema SPC (Sistema Pictográfico de Comunicación) y el sistema ARASAAC, desarrollado por este propio Portal Aragonés de CAA y que es de libre disposición con licencia Creative Commons.

5.4.2 Productos de apoyo para la comunicación

Pueden dividirse los productos de apoyo para la comunicación en básicos y tecnológicos. Los tableros de comunicación son productos de apoyo básicos que consisten en superficies de materiales diversos en las que se disponen los símbolos gráficos para la comunicación

(fotografías, pictogramas, letras, palabras y/o frases) que la persona indicará para comunicarse. Cuando los símbolos se distribuyen en varias páginas se habla de libros de comunicación.

Entre los productos tecnológicos se encuentran los comunicadores electrónicos especialmente diseñados para tal fin y los ordenadores portátiles o las tablets con programas especiales que los convierten en comunicadores. Los comunicadores electrónicos dedicados o emulados en ordenadores se personalizan con los símbolos gráficos que requiere cada persona y se caracterizan por ser portátiles y adaptarse a las formas de acceso apropiadas para cada persona (teclados, ratones, conmutadores, etc.). Disponen de una salida para los mensajes en forma de habla digitalizada o sintetizada, así como también, a menudo, de otras salidas como pantalla, papel impreso o incluso funciones de control del entorno.

5.4.3 Estrategias y productos de apoyo para el acceso

Gracias a las diferentes estrategias y productos de apoyo para el acceso, por muy restringida que se encuentre la movilidad de una persona, casi siempre es posible encontrar una solución para que pueda acceder a la comunicación, así como a otras actividades, tales como la movilidad asistida, el control del entorno o el acceso al ordenador para la escritura, el dibujo, el juego o la comunicación a través de la red.

Para indicar los símbolos gráficos en los comunicadores, tableros y libros de comunicación existen cinco estrategias fundamentales:

- **La selección directa.** Consiste en señalar o pulsar las teclas directamente, con el dedo, con la mirada o con otras partes del cuerpo, para indicar los pictogramas, palabras o letras que se quieren comunicar. Los punteros de distinto tipo son ejemplos de productos de apoyo que puede facilitar la selección o acceso directo.

- **La selección con ratón.** Solamente para productos electrónicos, consiste en acceder con un ratón a teclados o cuadrículas con símbolos para la comunicación en pantalla. Se pueden usar una gran variedad de ratones adaptados, en forma de joystick, trackball, así como el ratón facial (controlado con movimientos de la cabeza), el ratón controlado con la mirada o el multimouse, consistente en cinco teclas o conmutadores.
- **La exploración o barrido dependiente.** Solamente en tableros o libros, consiste en que el interlocutor vaya señalando, uno en uno o por grupos, filas y columnas, los símbolos o letras a comunicar, hasta que el hablante asistido indique con un gesto que se ha dado con el que quería comunicar.
- **La exploración o barrido independiente.** Solamente para productos electrónicos, en este caso es el comunicador u ordenador el que presenta las diferentes opciones a comunicar hasta que el hablante asistido selecciona la que le conviene pulsando un conmutador. Existen muchos tipos de conmutadores que se pueden activar con diferentes partes del cuerpo.
- **La selección codificada.** En este caso, cada símbolo o letra tiene un código, por ejemplo un número de dos o tres cifras o un color y un número, de manera que el hablante asistido indica de forma directa o por barrido este código para transmitir el símbolo o letra. De esta forma con pocas teclas o casillas puede acceder a un gran número de símbolos.

5.5 Juguetes adaptados

Se consideran juguetes adaptados aquellos juguetes estándar a los que se les ha hecho alguna modificación física que facilita el uso y el juego a un niño con discapacidad.

Estos consisten en un juguete con una o dos entradas de conmutador para poder ser accionado mediante pulsación. Hay muchos tipos de adaptaciones, desde muy sencillas, como poner velcro a fin de fijar el juego o juguete, hasta otras bastante más complicadas, como traducir los efectos sonoros a visuales o instalar un *jack* que facilite el uso del juguete.

Los criterios básicos que marcarán el tipo y grado de adaptación del juguete son la capacidad de movimiento del niño o niña, sus capacidades de manipulación y sus posibilidades de percepción sensorial. Las modificaciones que se apliquen deberán estar también en consonancia con la complejidad física, el grado de discapacidad y el nivel de comprensión del niño.

Ante cualquier tipo de adaptación hay que velar por las condiciones de seguridad y asegurarse de que las adaptaciones que se introduzcan no añadan ningún peligro para el niño o niña que lo usará o para los de su entorno.

Otro aspecto que debe tenerse en cuenta es el aspecto final del juguete una vez adaptado, velando para que éste siga siendo atractivo y deseable a los ojos de un niño con o sin discapacidad.

Actualmente la gama de juguetes adaptados no es muy amplia, ya que existen pocas empresas encargadas de dicha adaptación, los precios aumentan entre un treinta y un cuarenta por ciento respecto del juguete sin adaptar y no todos los juguetes son adaptables para los diferentes tipos y grados de discapacidad.

A continuación se describen dos juguetes, en forma de pack, que han sido adaptados:

- **BJ-K02, Kit de Juego de Bolos.** Consiste en un juego de bolos infantil ligero, el cual incluye un conmutador el cual puede accionar una sopladora eléctrica, también incluida y que hará que se mueva la bola. Y además, incluye un comunicador Go Talk, con 20 plantillas con pictogramas relacionados con la actividad.



Figura 5.1: BJ-K02, Kit de Juego de Bolos

- **Kit pecera adaptada.** Kit que permite dar alimentos a los peces de la pecera mediante un brazo articulado y un alimentador adaptado, además incluye un pulsador, una cámara de fotos adaptada y además incluye un comunicador Go Talk, con 20 plantillas con pictogramas relacionados con la actividad.



Figura 5.2: Kit pecera adaptada

5.6 Accesibilidad orientada a dispositivos móviles

Actualmente existen mecanismos que posibilitan la accesibilidad en dispositivos para personas con algunos tipos de discapacidad. Es un derecho, que todos tenemos y por ello, es necesario que se sigan desarrollando para que cualquier usuario, con discapacidad o no tenga las mismas oportunidades de interactuar con dispositivos móviles.

Hoy en día, en el mercado existen soluciones para personas con discapacidades auditivas y visuales, también se han lanzado otros dispositivos que facilitan las funciones a otros usuarios que, por cuestión de edad o algún otro tipo de problema físico no pueden tener fácil acceso al móvil.

5.6.1 Accesibilidad para personas con discapacidad auditiva

Las personas que tienen una discapacidad auditiva pueden utilizar un dispositivo móvil que sea compatible con un accesorio adicional llamado “lazo de inducción”.

El lazo de inducción es un filtro de ruidos que ayuda a quienes tienen un audífono con bobina telefónica "T" (es un audífono especial que funciona con teléfonos) a eliminar las interferencias para mantener una conversación.

Este accesorio se cuelga en el cuello y permite contestar llamadas con una sola tecla desde el dispositivo, así como también es posible utilizarlo como manos libres. Con el lazo de inducción también se puede apagar o encender el dispositivo móvil.

5.6.2 Accesibilidad para personas con discapacidad visual

Para las personas con discapacidad visual se usan soluciones a partir de diferentes programas, los cuales traducen el texto del menú de la pantalla y de los mensajes en voz. En

España los fabricantes de dispositivos móviles junto con la empresa Code Factory y la ONCE-CIDAT comercializan una aplicación diseñada para dispositivos móviles llamada Mobile Speak.

Mobile Speak funciona como un lector de pantalla para reconocer el texto y toda la información visual que aparece en el terminal y transformarla en sonido.

Esta aplicación hace posible que las personas con discapacidad visual puedan acceder a las funcionalidades más utilizadas en los dispositivos móviles. Hacer y recibir llamadas, mensajes de texto, registro de llamada y lista de contactos; navegar por internet, escuchar archivos de audio o conectarse con otros dispositivos por medio de Bluetooth o Wifi, son algunas de las funcionalidades que esta aplicación puede realizar.

5.6.3 Accesibilidad para personas mayores

Para las personas mayores o con alguna dificultad física o intelectual, la empresa austriaca Emporia comercializa dispositivos móviles pensados para utilizarse de manera sencilla, incluyendo en el terminal las funciones más básicas, como acceso a llamadas, mensajes de texto, agenda de contactos, entre otras cosas.

Dentro de la gama Emporia, existe uno de sus primeros dispositivos posee una gran pantalla con iluminación naranja para mejorar el contraste con el contenido, así como un volumen de timbre y altavoz adecuados para personas mayores, de forma que puedan escuchar con claridad sin que el sonido sea molesto. También incluye un botón rojo en la parte de arriba del dispositivo, en el cual se puede programar hasta cinco números de emergencia para tener acceso directo a ellos.

Actualmente la gama de teléfonos Emporia cuentan con servicios como llamada de emergencia con la que automáticamente se localiza la posición del usuario, gracias al GPS integrado. Cuentan con un nivel de vibración extrafuerte, botones de gran tamaño, un sensor de caídas o un reloj de pulsera.

Capítulo 6: Dispositivos externos de manejo del juguete

Durante las próximas líneas, el lector podrá encontrar una serie de objetos externos que pueden llegar a conectarse al dispositivo Android para el manejo de la aplicación u otras aplicaciones instaladas en el dispositivo Android. Se trata de un estudio teórico, por lo que las características de los diferentes dispositivos aquí escritas dependen de los fabricantes de cada uno de ellos.

6. 1 Inconvenientes del uso de Tablets PC o Teléfonos inteligentes

El uso de dispositivos Tablet PC o teléfonos inteligentes puede ser complicado para personas de movilidad reducida, o sin capacidad para poder emitir los sonidos correctos de forma que la aplicación sobre sistema Android no ejecute las indicaciones deseadas.

Uno de los objetivos de este Proyecto Fin de Carrera era realizar un estudio teórico de dispositivos externos que pudieran conectarse a un dispositivo Tablet PC o un teléfono inteligente de forma que facilitara el manejo de los mismos, o más en concreto, la aplicación de manejo del juguete teledirigido.

6.2 Posibles soluciones

En función de las necesidades de cada persona, y haciendo una clasificación reducida de los tipos de discapacidades, en las próximas líneas se describirán varios objetos que pueden conectarse a un sistema Android para salvar los problemas de movilidad, visuales o de comunicación.

6.2.1 Joystick

Una palanca de mando, o Joystick, es un dispositivo de control de dos o tres ejes que usa desde videoconsolas u ordenadores personales hasta un trasbordador espacial, aviones de última generación, grúas de carga y portacontenedores, tractores u otra maquinaria pesada que dispongan de un software para el control de operaciones especiales.

En sus inicios, los joystick fueron utilizados para controlar los alerones y el plano de profundidad de una nave experimental. Presente en los primeros aviones, su origen mecánico no está claro. El primer modelo eléctrico fue inventado en 1944 en Alemania durante la Segunda Guerra Mundial, siendo del tipo de dos ejes. El mismo, era utilizado en aviones de guerra,

haciendo que el controlador del arma pudiera guiar los misiles contra los objetivos deseados por control radio.

Más adelante, estos dispositivos fueron incorporados a los ordenadores personales y a las videoconsolas, de forma que todos los movimientos que se producían en los videojuegos que se ejecutaban en ambos dispositivos, eran controlados mediante el joystick.

Por el otro lado, el Apple II introdujo el joystick analógico con conector de 9 pines, posteriormente adoptado por el IBM PC pero con conector de 15 pines. Sólo unos pocos ordenadores de 8 bits utilizaron su propia variación del joystick analógico. Más tarde, la compañía japonesa Nintendo, con su modelo de videoconsola Nintendo Entertainment System, añadió la primera interfaz propietaria con señales multiplexadas, que con variaciones propietarias en cada consola y generación ha sido el sistema utilizado en las videoconsolas hasta la aparición del USB.

Por último, la compañía estadounidense Apple también desarrolló su propia interfaz, Apple Desktop Bus, como método de conexión de joysticks analógicos y digitales y gamepads en los Apple Macintosh (notablemente caros por las pocas cantidades fabricadas). Con la aparición de los modelos iMac se cambió la interfaz por USB.

Este invento fue incorporado para adelante para la ayuda a los discapacitados. Es fácilmente visible en las sillas de ruedas motorizadas. A día de hoy, mediante interfaz USB, se pueden utilizar ciertos tipos de joystick como sustitutos del ratón como dispositivo apuntador, para personas afectadas por parálisis cerebral, distrofia muscular, esclerosis lateral amiotrófica o artrogriposis congénita.

Los elementos de un joystick son los siguientes:

- Mango (1)
- Base (2)
- Botón de disparo (3)
- Botones adicionales (4)
- Interruptor de autodisparo (5)

- Palanca (6)
- Botón direccional (7)
- Ventosa (8)

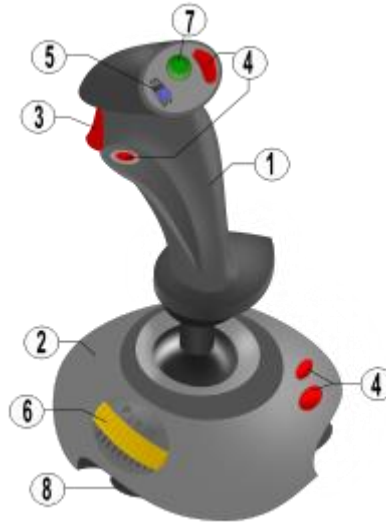


Figura 6.1: Joystick con las partes descritas anteriormente.

Con respecto a la aplicación que maneja el juguete teledirigido de este Proyecto Fin de Carrera, sí existen algunos modelos en el mercado, tanto de Tablet PC como de joystick, que pueden combinarse, de forma que el manejo de la Tablet puede realizarse, en gran parte, desde el joystick.

Un caso concreto es el modelo ASUS Eee Pad Transformer, que básicamente es un Tablet ARM, al que se le puede conectar a un accesorio aparte que lo transforma en notebook, tiene esa posibilidad tanto con los típicos mandos de PlayStation 3 y XBOX 360. En este caso, Android también ha puesto de su parte para que esto sea posible, ya que, a partir de la versión 3.1 permite conectar más dispositivos externos mediante USB.



Figura 6.2: ASUS con Joystick

Otra opción es el uso de USB On-the-Go, conocido como USB OTG o USB Host. Extensión de la norma USB 2.0, este invento permite flexibilidad en la gestión de la conexión USB.



Figura 6.3: USB-OTG

Gracias a esto, se puede conectar un amplio rango de periféricos al dispositivo Android, siempre y cuando el terminal soporte esta opción. De esta forma, se pueden conectar joystick populares como los utilizados en las videoconsolas para el manejo de aplicaciones en el dispositivo Android.



Figura 6.4: Ejemplos de utilización de USB-OTG

Por otro lado, el fabricante NGS tiene su modelo Sonar, el cual es un joystick para Tablet PC. En este caso, su conexión a la Tablet PC es diferente, ya que lo hace a través de Bluetooth, sujetándose gracias a la ventosa que lleva en su parte inferior sobre la pantalla táctil.



Figura 6.5: NGS Sonar

Siguiendo el ejemplo anterior, el fabricante Logitech tiene un joystick que sigue el sistema de ventosas sobre la pantalla táctil.

Para concluir, como último ejemplo de este tipo de componentes, también de conexión vía Bluetooth, se encuentra el modelo Gamepad de Samsung, el cual es exclusivo para la marca surcoreana.



Figura 6.6: Samsung Galaxy 4 con Samsung Gamepad

Con respecto a estos últimos ejemplos de conexión vía Bluetooth, conviene realizar una matización: teniendo en cuenta que la conexión por Bluetooth es única, y considerando que el Sphero es sincronizado por la misma vía, existen dos posibilidades de cara al futuro para llevar a cabo estas posibilidades:

- Sphero – WiFi: si el Sphero en un futuro puede sincronizarse vía WiFi en vez de Bluetooth, se podrían utilizar los mandos vía Bluetooth anteriormente citados.
- Bluetooth multiconexión: a día de hoy se puede realizar una múltiple conexión de un dispositivo con varios vía Bluetooth a modo de host. Los resultados no son los esperados en este caso, ya que una conexión tiene que dejar paso a la otra, y viceversa. El manejo de la aplicación sería posible, recibiendo las órdenes del joystick, pero al perderse la sincronización con el Sphero, hasta poder enviarlas pasaría el tiempo necesario para dicho efecto. Como se indica, los resultados no son los esperados, aunque en un futuro cercano podría llevarse a cabo.

6.2.2 Micrófono

Otro de los dispositivos que pueden facilitar el manejo del dispositivo Tablet PC es un micrófono. Es un sistema fácilmente localizable en cualquier tienda de electrónica. Gracias a él, una persona de movilidad reducida, puede comunicarse con su Tablet PC para que esta ejecute las instrucciones solicitadas.

En la actualidad, cualquier dispositivo Android dispone de entrada de audio. Aún así, por diferentes razones, ya sea comodidad o dificultad a la hora de emitir sonidos, un micrófono puede facilitar el uso de las aplicaciones del sistema Android.



Figura 6.7: Micrófono para Tablet PC o teléfonos inteligentes.

6.2.3 Google Glass

Uno de los inventos más innovadores de los últimos años ha sido, sin duda, Google Glass. Estas gafas son un dispositivo de visualización tipo gafas de realidad aumentada. Diseñadas por Google, la primera edición llamada Google Glass Explorer Edition fue lanzada en el año 2013, siendo el 15 de Abril de 2014 cuando salió la primera versión comercial de las mismas en Estados Unidos. En Europa fue posible obtenerlas a partir del 23 de Junio de ese mismo año.

El principal objetivo por parte de Google, con estas gafas, es hacer independiente el uso de los dispositivos Android como teléfonos inteligentes de las manos del usuario del mismo, permitiendo también el acceso a internet mediante órdenes de voz. En la actualidad, se puede encontrar una forma comparable a estas características en los dispositivos Android, gracias a Google Now.

Las principales características técnicas son las siguientes:

- No hay información oficial sobre la resolución de pantalla. Se ha sugerido la de 640 × 360 ya que se recomienda para los desarrolladores de aplicaciones.
- Cámara de 5 megapíxeles, capaz de grabar vídeo 720p
- Wi-Fi 802.11b/g
- Bluetooth
- Batería: hasta un día de uso, aunque indican que utilizarlas con Google Hangouts o para grabar vídeo reducirá la batería
- Almacenamiento de 16 GB sincronizados con la nube (12 GB disponibles)
- Texas Instruments OMAP 4430 SoC 1.2GHz Dual (ARMv7)
- 682MB RAM "prox"
- Giroscopio de 3 ejes.
- Acelerómetro 3 ejes
- Sensor geomagnético (brújula)
- Sensores de luz ambiente y sensor de proximidad interno.
- Sistema de inducción ósea para la transmisión del sonido. A partir de la versión 2.0 se permite la utilización de auriculares específicos.
- La montura es ajustable.
- Actualmente se basa en Android 4.0.4 modificado y adaptado al dispositivo



Figura 6.8: Google Glass

La comunicación entre las gafas Google Glass y el dispositivo Android, ya sea Tablet PC o teléfono inteligente, se realiza vía Bluetooth. Una vez establecida la sincronización entre ambos, desde las Google Glass se puede realizar y recibir llamadas telefónicas, recibir notificaciones de correo electrónico y mensajes SMS enviados al dispositivo Android, además de poder manejar otras aplicaciones instaladas en el dispositivo en cuestión.

Tal y como ocurría en los ejemplos de joystick vía Bluetooth, serían necesarios dos posibles cambios para el control del Sphero con la aplicación desarrollada: que el mismo fuese sincronizado a través de WiFi, o, en su defecto, que la sincronización entre dispositivo Android y Google Glass fuese, también, a través de WiFi.

6.2.4 Teclado para invidentes

En Marzo de 2006, un grupo de investigadores de la Universidad Autónoma de Barcelona (UAB) y de la Organización Nacional de Ciegos Españoles (ONCE) han desarrollado un teclado para PC que combina las teclas de funciones y desplazamiento de un teclado convencional con ocho teclas braille que permiten escribir en cualquier idioma.

Este teclado, que se puede conectar al PC por el puerto USB, está diseñado para facilitar las tareas informáticas a los invidentes habituados a las máquinas de escribir Braille.

Teniendo en cuenta que los dispositivos Tablet PC tienen entrada micro USB, simplemente adaptando la entrada del teclado (USB-OTG), tendríamos la posibilidad de manejar la Tablet PC mediante este teclado para invidentes.



Figura 6.9: Teclado USB para invidentes

6.2.5 Matamarcianos

Antes de la época actual de las videoconsolas, la única forma de poder acceder a un videojuego, más allá de los primeros para PC, eran las máquinas recreativas que se podían encontrar en bares, cafeterías o salones recreativos.

Dichas máquinas estaban formadas por un mueble de gran tamaño, donde en su interior, se colocaba un monitor para poder ver el videojuego en cuestión. Para el manejo del mismo, se añadían dos joystick (para la posibilidad de dos jugadores) y unos botones de mando.

Gracias a gente que añora aquella época, y aprovechando el potencial de las Tablets, se han creado unos muebles similares, pero de menor tamaño, de forma que, introduciendo el dispositivo Tablet PC en dicha caja, se puede simular las antiguas partidas a los videojuegos de aquella época.

Para poder conectar el sistema Matamarcianos a nuestro dispositivo Tablet PC, solo es necesario disponer de un puerto USB, cosa que de la que dispone cualquier Tablet PC con sistema operativo Android en el mercado.

Existen dos tipo de modelos, ambos personalizables en el aspecto exterior: uno para Tablets PC de 7 pulgadas, que disponen de cuatro botones, y otro para Tablets PC de mayor tamaño, que disponen de seis botones.

De esta forma, y pensando de cara al futuro, la aplicación para el manejo del juguete teledirigido de este Proyecto Fin de Carrera, u otras aplicaciones para discapacitados, pueden controlarse de una forma más sencilla mediante un Joystick y unos botones externos.



Figura 6.10: Matamarcianos personalizado con el videojuego Space Invaders

Capítulo 7: Aplicación para la accesibilidad Sphero

Durante las próximas páginas el lector podrá ver todo lo relacionado con la aplicación para la accesibilidad Sphero, desde una descripción del diseño hasta su implementación, mostrando diagramas, imágenes y fragmentos de código para la correcta comprensión de todo lo desarrollado.

7.1 Diseño

7.1.1 Especificación de casos de uso

Los casos de uso son la representación gráfica de las funcionalidades de la aplicación. Los personajes o entidades que aparecen se denominan actores. Dentro de la Ingeniería del Software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

En este caso, se hará una comparativa, de la primera versión con esta segunda. Para la primera, el gráfico que nos encontramos es el siguiente:

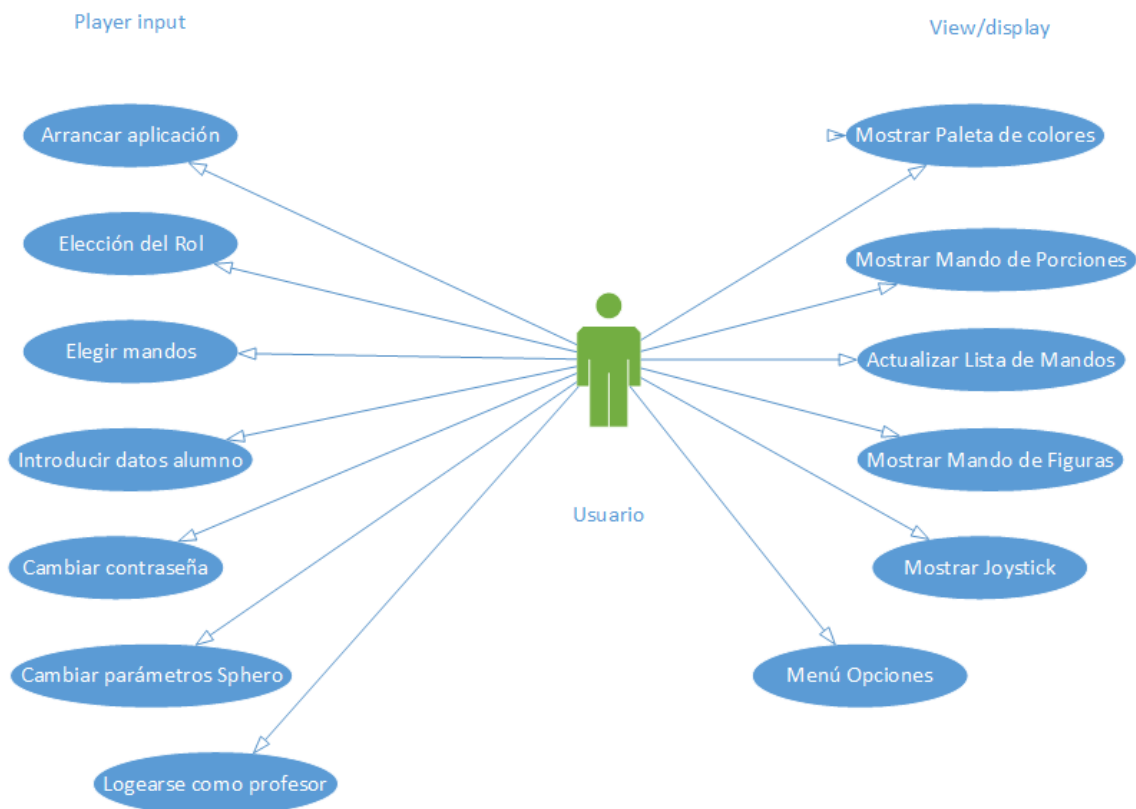


Figura 7.1: Casos de uso de la primera versión

En el diagrama presentado, cada caso de uso consiste en un óvalo con un nombre descriptivo en su interior. Además se encuentra un actor, que es el que va a interactuar con todo el sistema. Existe un único actor de nombre “usuario” que es el que realiza todos los casos de uso.

A continuación, se ofrece una breve descripción del cometido de cada uno de los casos de uso mostrados en el diagrama:

- **Arrancar aplicación:** consiste en la ejecución del programa principal.
- **Elección de rol:** escoger entre rol de profesor o alumno.
- **Elegir mandos:** escoger entre una lista de mandos, los que el usuario crea convenientes.
- **Introducir datos del alumno:** escribir los datos necesarios para identificar al alumno que va a utilizar los mandos y así particularizar el comportamiento realizado.
- **Cambiar contraseña:** introducir una contraseña diferente a la existente, por motivos de seguridad.
- **Cambiar parámetros Sphero:** introducir una serie de valores que harán que el funcionamiento de Sphero sea diferente al que viene por defecto, y favorecerán su manejo.
- **Logearse como profesor:** consiste en introducir las credenciales pedidas para acceder al menú del usuario con rol profesor.

Para esta segunda versión, la situación gráfica sería la siguiente:

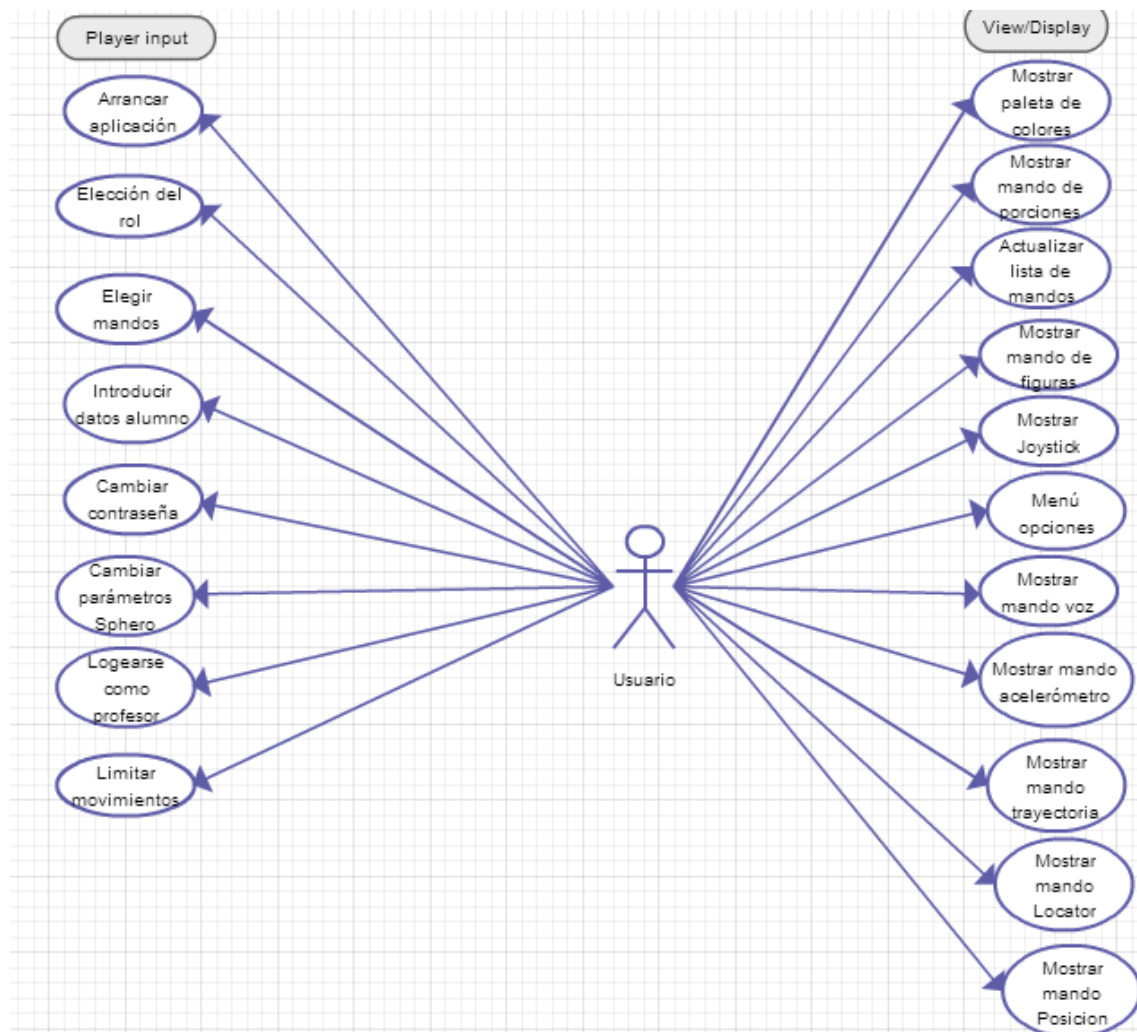


Figura 7.2: Casos de uso de la segunda versión

A continuación, se ofrece una breve descripción del cometido de cada uno de los casos de uso añadidos para esta segunda versión:

- **Introducir factores de corrección:** para uno de los nuevos mandos desarrollados, es necesario que el profesor indique que movimientos permite realizar al alumno, además de introducir los ángulos de corrección de los mismos.

- **Activar Bluetooth:** siguiendo la aplicación oficial del fabricante Orbotix, cuando el Bluetooth no está activado en la Tablet PC, aparece un icono del mismo que abre los ajustes de Android para que se pueda activar manualmente y sincronizar la Tablet PC con el juguete robótico Sphero.

7.1.2 Interfaz de usuario

Dentro de las herramientas existentes en el mercado para el desarrollo de software, y más concreto, aplicaciones Java para Android, está la más habitual que es Eclipse. Para ello, solo es necesario una versión IDE de Eclipse con una función ADT (Android Developer Tools) e instalar el SDK de Android. El SDK de Android proporciona las bibliotecas API y las herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android.

Aun siendo más popular Eclipse, para la realización de este proyecto se ha tomado la decisión de realizarlo con Android Studio. La principal razón ha sido ampliar el conocimiento con una herramienta no habitual, con la que puede aportar un poco más de aprendizaje de este proyecto.



Figura 7.3: Android Studio arrancando

Para la realización del aspecto gráfico, se ha desestimado la opción de realizarlo con lenguaje de programación Java, siguiendo las pautas de la primera versión. Como se indica en la misma, es algo posible y viable, pero es infinitamente más costoso, además de desaprovechar el modelo vista-controlador que ofrece Android. Así, es posible dibujar el aspecto gráfico mediante ficheros xml, implementando, a su vez, la parte lógica mediante lenguaje de programación Java.

7.2 Implementación

Antes de comenzar la implementación, se realizó un estudio de la viabilidad de los posibles desarrollos a realizar, propuestos en la primera versión, más una serie de mejoras que podrían completar de forma exitosa esta segunda versión. Una vez completado el análisis, se comenzó con la implementación de todas las ideas propuestas, adaptándolas a las posibilidades que tanto Android como la API del juguete robótico Sphero ofrecen.

Para conseguir una mayor comprensión por parte del lector, se han incluido partes del código en las próximas páginas.

7.2.1 Lenguaje de programación y entorno de desarrollo

Para el desarrollo de esta aplicación sobre Android se ha elegido el lenguaje de programación Java, aunque el sistema posee librerías desarrolladas en C/++. Una vez decidido el lenguaje de programación, se han elegido estos parámetros de Android para realizar la versión:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="main.actividad1"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
```

Figura 7.4: Parámetros SDK Android

Como se ha comentado en apartados anteriores, es necesario descargar el SDK de Android para la realización del desarrollo. Google, como propietario de Android, pone a disposición a los programadores, de forma gratuita, la descarga del SDK Android. Recordemos que el SDK contiene las bibliotecas básicas para utilizar algunas funciones que contiene la plataforma y que dan acceso al hardware.

Por último, tal y como se ha comentado en el punto anterior, se ha decidido utilizar Android Studio como entorno integrado de desarrollo. Esta es la versión instalada para realizar el desarrollo de la aplicación:



Figura 7.5: Versión Android Studio

7.2.1.1 Configuración del entorno de desarrollo

Antes de comenzar el desarrollo, es importante preparar el entorno para realizar el mismo. Dentro de las próximas líneas, se hablará de la configuración de los siguientes elementos: JDK Java, IDE Android Studio y SDK Android:

- **Instalación de Java Platform (JDK) desde la web de Oracle.** Una de las condiciones básicas es tener instalada la correcta versión de Java Platform en el equipo de desarrollo.

De no tenerlo, se puede obtener gratuitamente en la web de Oracle, pudiendo obtener la versión adecuada para cada equipo.



Figura 7.6: Enlace de descarga de JDK

En la imagen anterior se muestra la última versión disponible. Antes de redactar este documento, la última versión disponible era la 7.67 de Java 7 y la 11 de Java 8. Con la que se ha realizado todo el desarrollo:



Figura 7.7: Versión de Java con la que se ha desarrollado el proyecto

- **Instalación del IDE Android Studio.** Una vez instalado el JDK, se procederá a la instalación de Android Studio, desde su propia web. Para ello, accedemos a la web de Android Developers, y descargamos, de forma gratuita, la versión existente del entorno integrado de desarrollo propio del fabricante de Android.

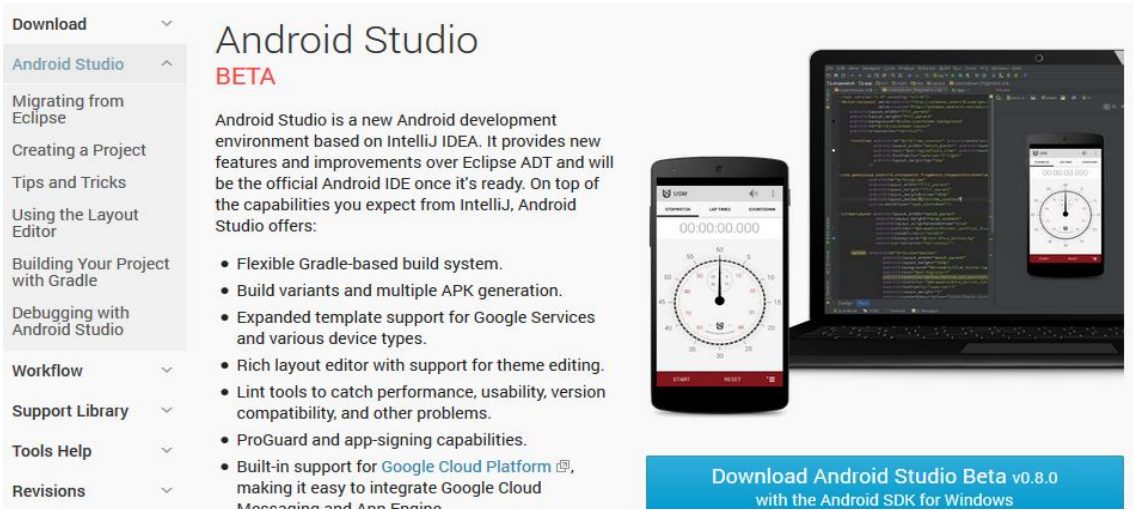


Figura 7.8: Descarga de Android Studio

En la web de Android Developers se puede encontrar una comparativa entre Android Studio y el ADT de Eclipse:

| Feature | Android Studio | ADT |
|---|------------------------|---------------------|
| Build system | Gradle | Ant |
| Maven-based build dependencies | Yes | No |
| Build variants and multiple-APK generation (great for Android Wear) | Yes | No |
| Advanced Android code completion and refactoring | Yes | No |
| Graphical layout editor | Yes | Yes |
| APK signing and keystore management | Yes | Yes |
| NDK support | Coming soon | Yes |

Figura 7.9: Comparativa Android Studio y ADT

Además, se pueden ver las diferentes versiones existentes del producto. En este caso, se ha realizado el desarrollo con la versión v.0.5.2, publicada en mayo de 2014:

- ▶ [Android Studio v0.8.0 \(June 2014\)](#)
- ▶ [Android Studio v0.5.2 \(May 2014\)](#)
- ▶ [Android Studio v0.4.6 \(March 2014\)](#)
- ▶ [Android Studio v0.4.2 \(Jan 2014\)](#)
- ▶ [Android Studio v0.3.2 \(Oct 2013\)](#)
- ▶ [Android Studio v0.2.x \(July 2013\)](#)
- ▶ [Android Studio v0.1.x \(May 2013\)](#)

Figura 7.10: Versiones de Android Studio

- **Instalación SDK Android.** A diferencia de Eclipse, con la instalación de Android Studio no es necesario este paso. De haber trabajado con Eclipse, hay que descargarlo de la web de Android Developers, instalarlo e integrarlo dentro del IDE Eclipse.

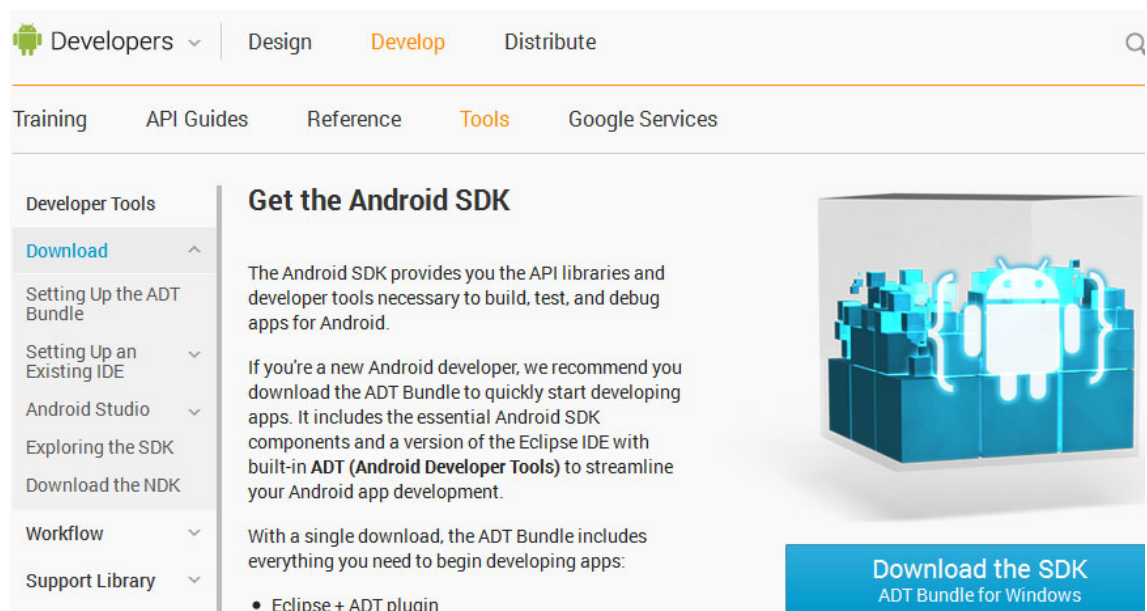


Figura 7.11: Enlace de descarga de SDK Android

7.2.1.2 Integración del SDK Sphero en el proyecto de Android Studio

Una vez completada la preparación del entorno de desarrollo, si se desea trabajar con unos métodos y atributos del fabricante Orbotix, propietario del juguete robótico Sphero, se debe descargar e integrar el SDK de Sphero en el proyecto creado en Android Studio. Para ello, se deben seguir los siguientes pasos:

- Acceder a la web de Sphero, propiedad de Orbotix, y dentro del enlace Developers, descargar el SDK Sphero. Existen varios tipos de SDK, pero en este caso, se debe acceder a la información relacionada con Android:

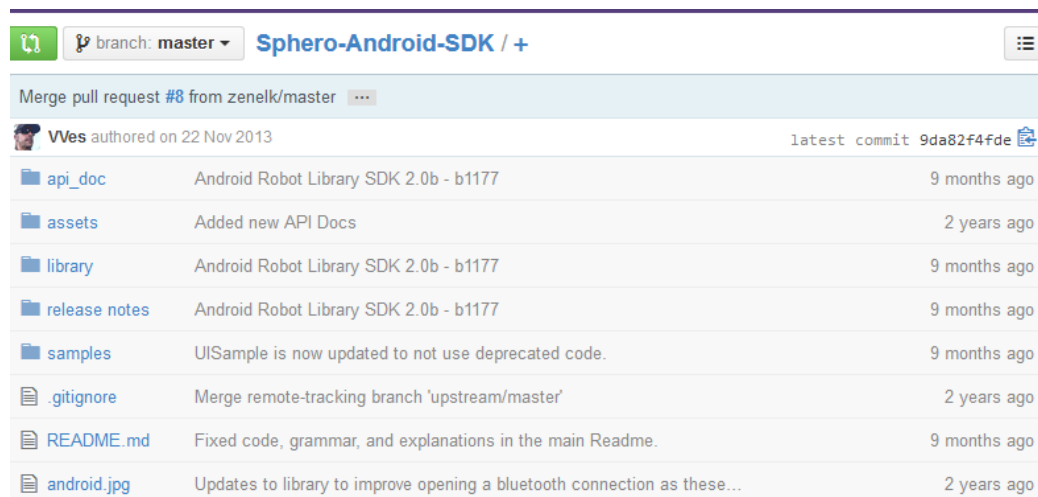


Figura 7.12: Sphero Android SDK

- Una vez descargado, se deberán copiar las librerías que se proporcionan, en nuestro proyecto de Android Studio:

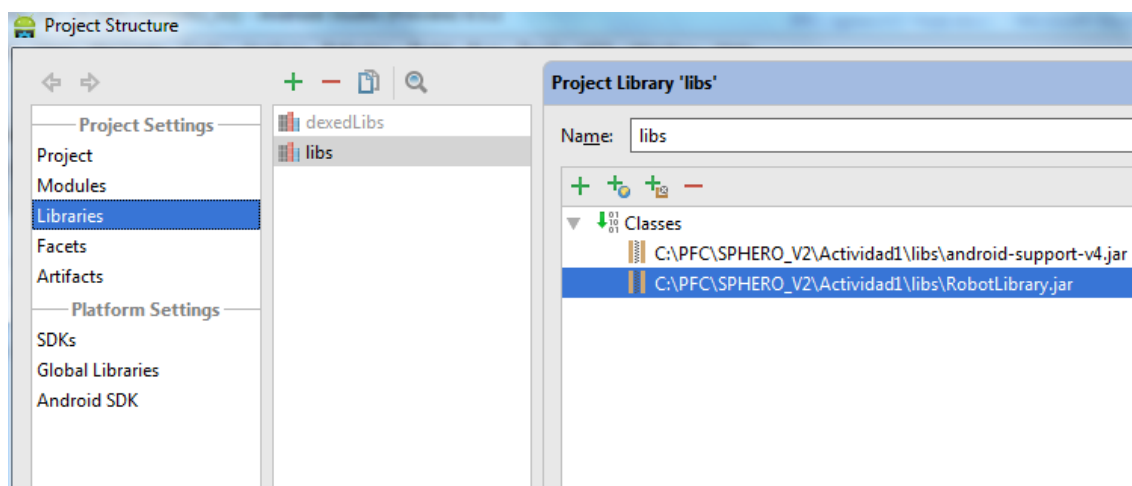


Figura 7.13: Añadido de librerías en Android Studio

El fichero marcado en la imagen anterior es el jar proporcionado por Orbotix para acceder al SDK de Sphero.

- Una vez realizados los puntos anteriores, se debe comprobar que están referenciadas las librerías de Orbotix en el buildpath del proyecto. Si así es, se puede comenzar a utilizar todos los atributos y métodos para poder manejar el juguete robótico Sphero.

7.2.1.3 Integración del SDK Flurry en el proyecto de Android Studio

Tal y como se comenta en el capítulo 4 de este documento, se ha tomado la decisión de eliminar todo lo relacionado con el producto Flurry Analytics. Para ello, y como se puede comprobar en la imagen de la integración del SDK de Sphero en Android Studio, se ha procedido a eliminar el fichero jar integrado en la primera versión.

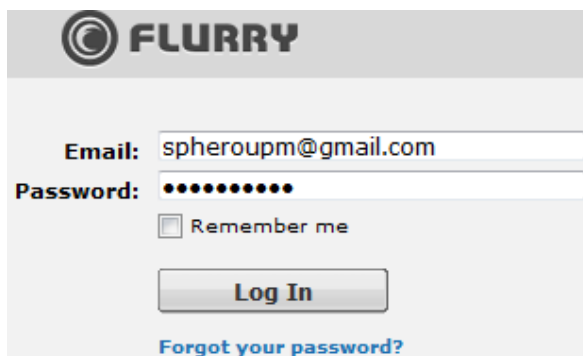


Figura 7.14: Pantalla de login en la web de Flurry

A partir de esta versión, se utilizarán los Webservices de Adaptare para obtener la información de las estadísticas de uso de la aplicación de control remoto del Sphero.

7.2.2 Realización del Proyecto

Si el lector ha tenido la oportunidad de leer la documentación de la primera versión, podrá recordar que se escribían estas líneas acerca de la doble finalidad del proyecto:

- Por un lado, es una aplicación que promueve la interacción, de usuarios con discapacidad, con un dispositivo externo radiocontrol; además favorece la integración del uso actual de dispositivos móviles con este tipo de usuarios. Es importante remarcar, que a día de hoy, existen un grupo muy reducido de aplicaciones orientadas a la discapacidad. Bien por temas económicos, ya que, para realizar una aplicación de este tipo requiere seguir unos estándares de diseño y un comportamiento determinado y todo ello conlleva a utilizar un desarrollador específico, que conozca estas pautas a la hora de programar. Además conlleva un riesgo, y es que no todas las aplicaciones orientadas a la accesibilidad son manejables por cualquier usuario, hay que acotar un grado y el tipo de discapacidad. Ello conlleva a reducir enormemente el número de usuarios para la que va a ir destinada, y para muchas empresas de desarrollo de software, esto es algo poco viable, ya que los ingresos van a ser menores, que una aplicación que va a ir destinada a cualquier sector de población. Con ello quiero remarcar que la primera finalidad es ofrecer la posibilidad de usar un juguete, como Sphero, que está orientado a un sector de población sin discapacidad, haciendo uso de cualquier dispositivo móvil bajo sistema operativo Android, a personas con diferentes grados de discapacidad sensorial, motora y psíquica.
- Por otro lado, esta aplicación sirve para obtener resultados una vez que ella sido utilizada y que van a ayudar a beneficiar de forma terapéutica al usuario que ha hecho uso de ella. En estos resultados, van a ir incluidos todos los eventos que se producen al pulsar botones, tocar diferentes zonas de la pantalla, cambiar de mandos, etc. Y que van a

posibilitar a una persona especializada en el tratamiento de personas con discapacidad, llevar un control del uso que hace el paciente y ver posibles mejoras que refleja el paciente a lo largo del tiempo.

Tomando como referencia estos dos puntos, y teniendo en cuenta cuales eran los objetivos de esta segunda versión, estas son las finalidades que se podrían destacar a día de esta segunda versión:

- Uno de los problemas con los que se encontraba el usuario de la aplicación de la primera versión es el difícil control de la velocidad del Sphero en muchos casos. Si tenemos en cuenta que la aplicación está destinada para gente discapacitada, ese problema puede convertirse en un gran inconveniente para el manejo del juguete robótico. Para esta nueva versión se han adaptado las velocidades para su mayor facilidad de manejo.
- Por otra parte, toda lo relacionado con la interacción entre usuario y Sphero pasaba por el control manual del dispositivo Tablet PC. Muchas de las personas con movilidad reducida ya tienen problemas para manejar aplicaciones más sencillas dentro de un dispositivo Tablet PC. Si a eso le añadimos la dificultad del manejo del Sphero, otra vez, tenemos un gran inconveniente. Para solucionarlo, se han creado algunos mandos que faciliten la interacción con el dispositivo Tablet PC: control por voz, control por movimiento de la Tablet PC, o mandos más fáciles de manejar que el principal, debido a que el dibujo de una trayectoria puede ser más accesible que el manejo de un mando que simula a un joystick. Por la tanto, la finalidad de ampliar el sector de la población que pudiera interactuar con la Tablet PC para el manejo del Sphero se ha ampliado notablemente.
- Con respecto a los resultados en los que se habla en la primera versión, el producto Flurry Analytics no ha satisfecho las expectativas de su uso, debido a que los datos almacenados son genéricos y poco ayudan al análisis de los resultados del uso de la aplicación de control teledirigido de un juguete robótico. Por esta razón, se ha creado una aplicación específica para esta nueva versión, de forma que ahora el profesor o analista que desee, podrá sacar conclusiones del uso del juguete Sphero.

7.2.2.1 Clase Principal

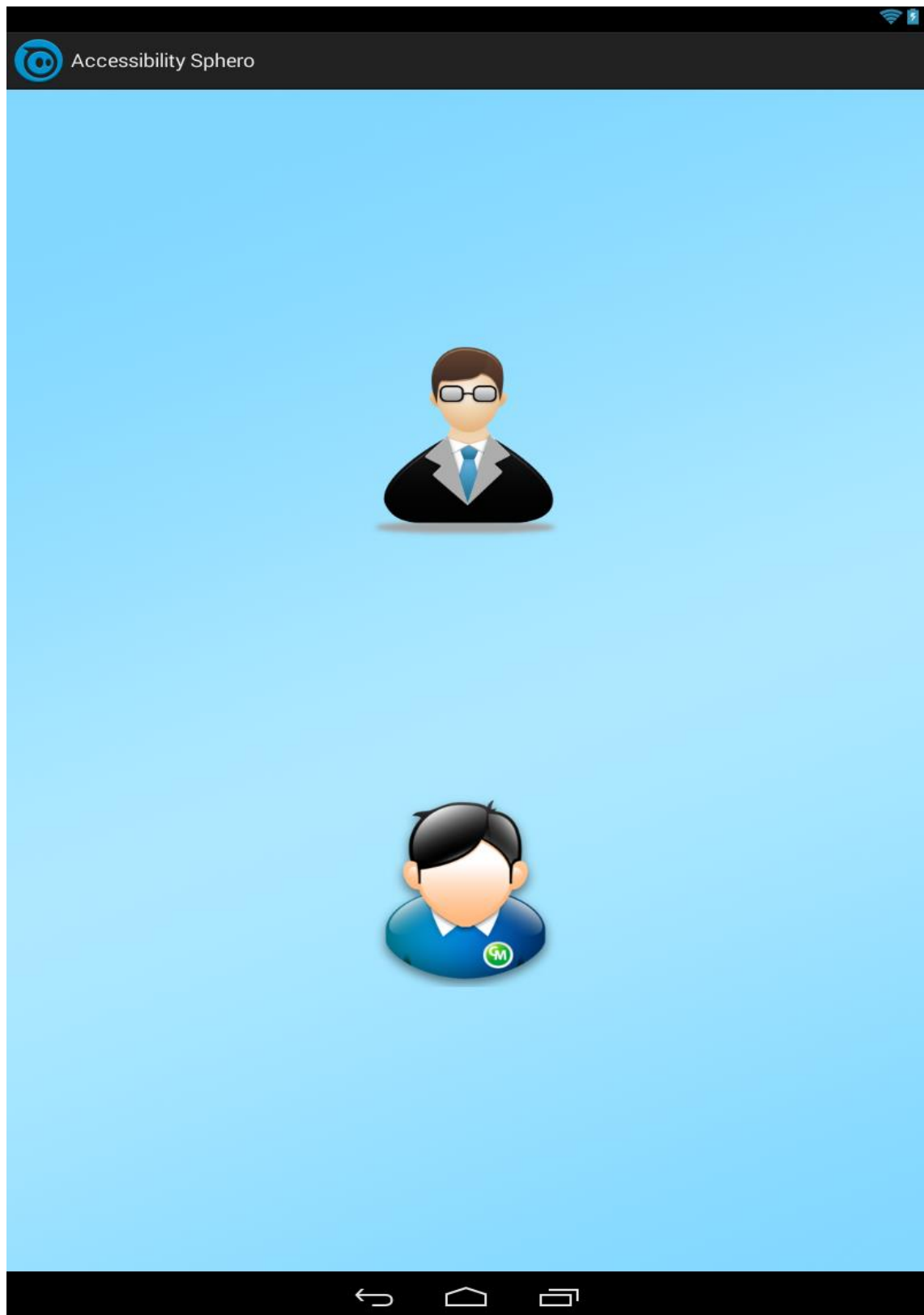


Figura 7.15: Pantalla correspondiente al Activity Main

Esta parte no ha sufrido cambios con respecto a la primera versión. Corresponde a la “Activity Main” y siguen manteniéndose los dos posibles roles: alumno y profesor. Una vez seleccionado uno de los dos posibles roles, se accederá a la actividad particular de cada uno de ellos. Con respecto a la parte visual, se mantiene el mismo Relative Layout con dos botones en los que se carga una imagen y se les asigna un método en el atributo onClick, a ejecutar cuando se marque uno de los dos roles que aparecen en pantalla.

La única parte alterada fue con respecto a dos errores aparecidos, que se comentarán en el próximo capítulo.

7.2.2.2 Clase de acceso al Perfil de profesor

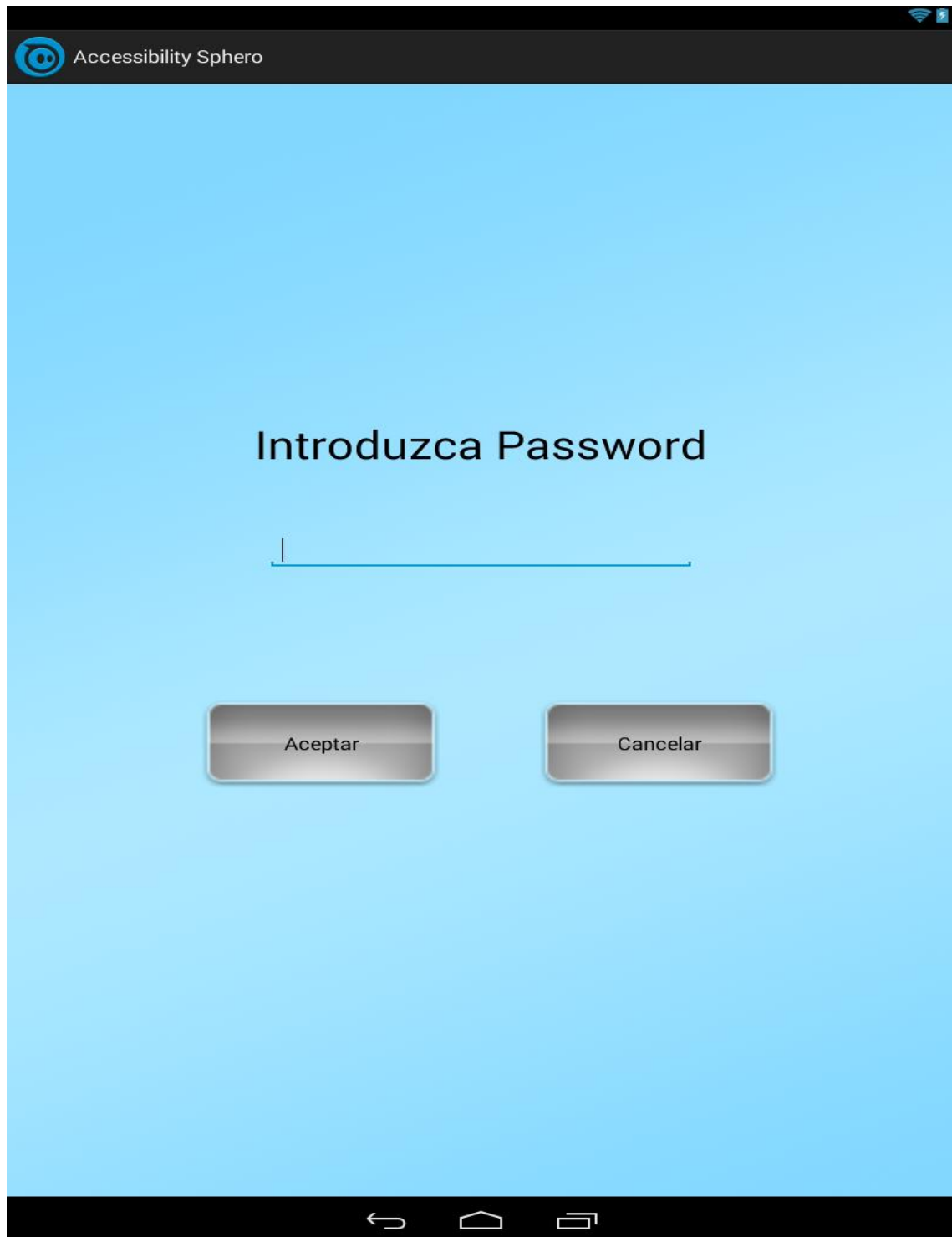


Figura 7.16: Pantalla correspondiente al Activity PasswordProfe

Al igual que la clase anterior, la clase de acceso al perfil del profesor tampoco ha sufrido cambios con respecto a la primera versión. La clase corresponde a la “Activity PasswordProfe”, a la cual se accede tras pulsar el icono del rol de profesor de la actividad precedente. Al solicitar una contraseña para el acceso, el profesor se asegura que el alumno no tenga posibilidad de entrar y modificar su perfil.

Toda la parte relacionada con la validación de la contraseña y el almacenamiento de la misma se ha mantenido tal y como se encontraba en la primera versión.

En la parte visual está realizada con un único Relative Layout, un Textview para mostrar la etiqueta “Introduzca Password”, un EditText para escribir la contraseña y dos botones en los que se carga una imagen y se les asigna un evento en el atributo onClick. En el caso de pulsar el botón “Aceptar”, si la contraseña es correcta, se avanzará a una nueva actividad, mientras que si el resultado es incorrecto, se mostrará un mensaje de error informando de la circunstancia, manteniéndose en esta actividad. Por último, en caso de elegir el botón “Cancelar”, la aplicación regresará a la actividad anterior.

7.2.2.3 Clase Menú del profesor



Figura 7.17: Pantalla correspondiente al Activity MenuProfesor

La clase corresponde a la “Activity MenuProfesor”, donde se accede tras validar la contraseña de acceso de forma correcta en la actividad anterior. Esta clase tampoco ha sufrido cambios con respecto a la versión anterior.

En esta actividad, el usuario que actúe bajo el rol de profesor, tiene dos posibles opciones:

- **Configuración:** accediendo al menú de configuración se pueden cambiar los parámetros relacionados con Sphero, con el alumno que vaya a utilizar la aplicación y la contraseña del menú del profesor. Además, se ha eliminado la parte de configuración del producto Flurry Analytics, y se ha incorporado el acceso a una nueva actividad para los factores de corrección de uno de los nuevos mandos, pero ya dentro de la actividad de los datos del alumno. Esta parte se describirá más adelante.
- **Selección de Mandos:** la funcionalidad de esta actividad tampoco ha sido modificada, pero si ha visto incrementar la colección de mandos disponibles a seleccionar por el usuario con rol de profesor y a manejar por el usuario con rol de alumno. Más adelante se comentarán los cambios.

Sobre la parte visual, siguiendo el modelo de las pantallas anteriores, solo está compuesta por un único Relative Layout, con dos botones en los que se carga una imagen y un método asociado en sus atributos onClick.

7.2.2.4 Clase Configuración avanzada



Figura 7.18: Pantalla correspondiente al Activity ConfigAvanzada

Si en el menú anterior el profesor se selecciona el icono de “Configuración”, aparecerá esta ventana. La clase corresponde a la “Activity ConfigAvanzada”, y tampoco ha sido modificada para esta nueva versión, salvo algún mínimo detalle como el nombre de la aplicación: Adaptare Sphero.

Dentro de la actividad se pueden observar una serie de etiquetas con botones al lado, los cuales accederán a una nueva actividad o simplemente modificarán los parámetros relacionados. He aquí una breve descripción de cada uno de ellos:

- **Datos Alumno:** posee un botón al lado con la etiqueta “Alumno”, el cual, si es pulsada, accederá a una nueva actividad que contiene los datos del alumno que va a utilizar la aplicación de control del juguete Sphero.
- **Contraseña:** permite cambiar la contraseña del profesor, accediendo a una nueva actividad. La parte relacionada con el token de Flurry ha sido eliminada.
- **Iluminación de cola:** permite activar o desactivar la luz de cola (LEDTail) del Sphero, la cual marcará la dirección del juguete.
- **Límite de distancia:** si está activado, impedirá al Sphero alejarse a una distancia mayor de la que puede cubrir la conexión entre la Tablet PC y el juguete vía Bluetooth.
- **Retardo de inicio:** permite asignar un valor. Sirve para indicar un valor, en segundos, del tiempo de espera desde que envía un comando hasta cuando se ejecuta.
- **Tiempo Off:** permite asignar un valor. Dicho valor indica el tiempo máximo de espera sin acciones que puede llegar a estar el Sphero sin apagarse para ahorrar batería.

- **Vel. Máxima:** regula el valor de la máxima velocidad que puede alcanzar el Sphero. Mediante un Seekbar, se podrá establecer un máximo en el parámetro velocidad, aunque para las nuevas funcionalidades se ha decidido trabajar con velocidad fijadas desde un principio, para facilitar el manejo del juguete.

Para la parte visual, se ha contado con un Relative Layout con un fondo diferente en esta actividad, además de las que se acceden desde los botones Alumno y Contraseña. Además, se cuenta con dos ToogleButtons para las activaciones de iluminación de cola y límite de distancia, y, por último, varios TextViews para las etiquetas fijas y varios EditText para los valores a introducir manualmente.

Los valores aquí indicados se guardarán en un directorio interno del dispositivo Android, de forma que cuando se vuelva a iniciar la actividad, puedan ser rescatados y reutilizados, sin necesidad de estar configurando los datos del alumno en cada momento que vaya a utilizar la aplicación.

```
public void setParameters(String headerLight, String limDistance,
    String startDelay, String maxVel, String offTime) {

    File dir = new File(
        android.os.Environment
            .getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC),
        "Spheropassword");
    if (!dir.exists()) {
        dir.mkdirs();
    }
    String filename = "Parameters.txt";
    try {
        File f = new File(dir + File.separator + filename);
        FileWriter fileWriter = new FileWriter(f);
        BufferedWriter bufferWriter = new BufferedWriter(fileWriter);
        bufferWriter.write(headerLight);
        bufferWriter.write("-");
        bufferWriter.write(limDistance);
        bufferWriter.write("-");
        bufferWriter.write(startDelay);
        bufferWriter.write("-");
        bufferWriter.write(maxVel);
        bufferWriter.write("-");
        bufferWriter.write(offTime);
        bufferWriter.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 7.19: Almacenamiento de parámetros en fichero Parameters.txt

7.2.2.5 Clase Datos del alumno

Adaptare Sphero

Datos del Alumno

Nombre: _____

Fecha de Nacimiento:
(dd/mm/aaaa)

Género: ☐ Masculino ☐ Femenino

Discapacidad 1:

Item 1
Sub Item 1

Discapacidad 2:

Item 1
Sub Item 1

Discapacidad 3:

Item 1
Sub Item 1

Aceptar Cancelar Limitar

Figura 7.20: Pantalla correspondiente al Activity FlurryUserData

La clase corresponde a la “Activity FlurryUserData”, es a la cual se accede después de que el usuario haya pulsado el botón “Alumno” en la actividad anterior (ConfigAvanzada). Esta clase ha sido modificada, primero, para conservar los datos de Flurry y utilizarlos para Adaptare, y, segundo, incluir el botón de limitación de movimientos, con las correspondientes correcciones de los mismos. Además, se ha completado el perfil del alumno pudiendo elegir hasta tres tipos de discapacidades distintas.

Los datos del alumno, que pueden introducirse en esta pantalla, anteriormente se utilizaban para generar las estadísticas en la web de Flurry. Con el paso de Flurry a Adaptare, se aprovechan los mismos datos, ya que son los necesarios para realizar los estudios correspondientes. Además, se han ampliado para poder introducir hasta tres niveles de discapacidad, de forma que así los datos sean más válidos para realizar análisis sobre la actividad de la aplicación. Los datos que aquí aparecen son los siguientes:

- **Nombre del alumno:** un pseudónimo asociado al alumno.
- **Fecha de nacimiento:** como en estos casos, siempre es más preciso utilizar este dato que la edad.
- **Género del alumno:** masculino o femenino.
- **Primera discapacidad:** permite asociar la discapacidad con el alumno.
- **Segunda discapacidad:** a utilizar en el caso que fuese necesario.
- **Tercera discapacidad:** a utilizar en el caso que fuese necesario.

Completando esta parte, y debido a la inclusión del nuevo mando de movimientos limitados, se ha incluido el botón “Limitar”, de forma que iniciar una nueva actividad donde se pueden marcar los movimientos que puede realizar el alumno (se explicarán más adelante) y los datos de corrección de los mismos.

Por último, en la parte visual, se ha utilizado un Relative Layout con el mismo fondo que la actividad anterior, seis etiquetas Textview y cinco campos EditText donde el profesor puede rellenar los datos necesarios. Además, se han incluido dos Checkbox, donde se debe seleccionar la entrada del género, tres Spinners para los niveles de discapacidad y tres botones con imágenes, de Aceptar para modificar los datos, Cancelar para volver a la actividad anterior sin modificar los datos, y Limitar, que iniciará la actividad descrita más adelante de limitación de movimientos.

Todos los datos serán guardados en la memoria del dispositivo Android, de forma que cada vez que se vaya a utilizar la aplicación se conserven y puedan utilizarse de nuevo. Además, aprovechando la base de datos de Adaptare, serán guardados en la misma, de forma que pueden consultarse a través de la aplicación web con dicho nombre.

```
public void setParamsUserFlurry(String paramsFlurry) {  
  
    File dir = new File(  
        android.os.Environment  
            .getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC),  
        "Spheropassword");  
    if (!dir.exists()) {  
        dir.mkdirs();  
    }  
    String filename = "ParamsFlurry.txt";  
    try {  
        File f = new File(dir + File.separator + filename);  
  
        FileWriter fileWriter = new FileWriter(f);  
        BufferedWriter bufferWriter = new BufferedWriter(fileWriter);  
        bufferWriter.write(paramsFlurry);  
        bufferWriter.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Figura 7.21: Almacenamiento de datos de usuario en la Activity FlurryUserData

```
public String getParamsUserFlurry() {
    File dir = new File(
        android.os.Environment
            .getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC),
        "Spheropassword");
    String paramsFlurry = "";
    if (!dir.exists()) {
        dir.mkdirs();
    }
    String filename = "ParamsFlurry.txt";

    try {
        File f = new File(dir + File.separator + filename);
        FileReader fileReader = new FileReader(f);
        BufferedReader bufferedReader = new BufferedReader(fileReader);
        paramsFlurry = bufferedReader.readLine();
        String[] parameters = paramsFlurry.split("-");
        nomUsuario.setText(parameters[0]);
        fechaUsuario.setText(parameters[1]);
        if(parameters[2].equals("masculino")){
            generoMasculino.setChecked(true);
            generoFemenino.setEnabled(false);
        }
        else{
            generoFemenino.setChecked(true);
            generoMasculino.setEnabled(false);
        }

        bufferedReader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return paramsFlurry;
}
```

Figura 7.22: Obtención de datos de usuario en la Activity FlurryUserData

```
public void setParamsUserFlurryDiscapacidad(String paramsFlurry) {

    File dir = new File(
        android.os.Environment
            .getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC),
        "Spheropassword");
    if (!dir.exists()) {
        dir.mkdirs();
    }
    String filename = "ParamsFlurryD.txt";
    try {
        File f = new File(dir + File.separator + filename);

        FileWriter fileWriter = new FileWriter(f);
        BufferedWriter bufferWriter = new BufferedWriter(fileWriter);
        bufferWriter.write(paramsFlurry);
        bufferWriter.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 7.23: Nuevo método para almacenar las discapacidades en un fichero diferente


```
public String getParamsUserFlurryDiscapacidad() {  
    File dir = new File(  
        android.os.Environment  
            .getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC),  
        "Spheropassword");  
    String paramsFlurry = "";  
    if (!dir.exists()) {  
        dir.mkdirs();  
    }  
    String filename = "ParamsFlurryD.txt";  
  
    try {  
        File f = new File(dir + File.separator + filename);  
        FileReader fileReader = new FileReader(f);  
        BufferedReader bufferReader = new BufferedReader(fileReader);  
        paramsFlurry = bufferReader.readLine();  
        String[] parameters = paramsFlurry.split("-");  
  
        spinner1.setSelection(new Integer(parameters[0]).intValue());  
        spinner2.setSelection(new Integer(parameters[1]).intValue());  
        spinner3.setSelection(new Integer(parameters[2]).intValue());  
  
        bufferReader.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return paramsFlurry;  
}
```

Figura 7.24: Nuevo método para obtener las discapacidades en un fichero diferente

7.2.2.7 Clase *Limitar movimientos y factores de corrección.*

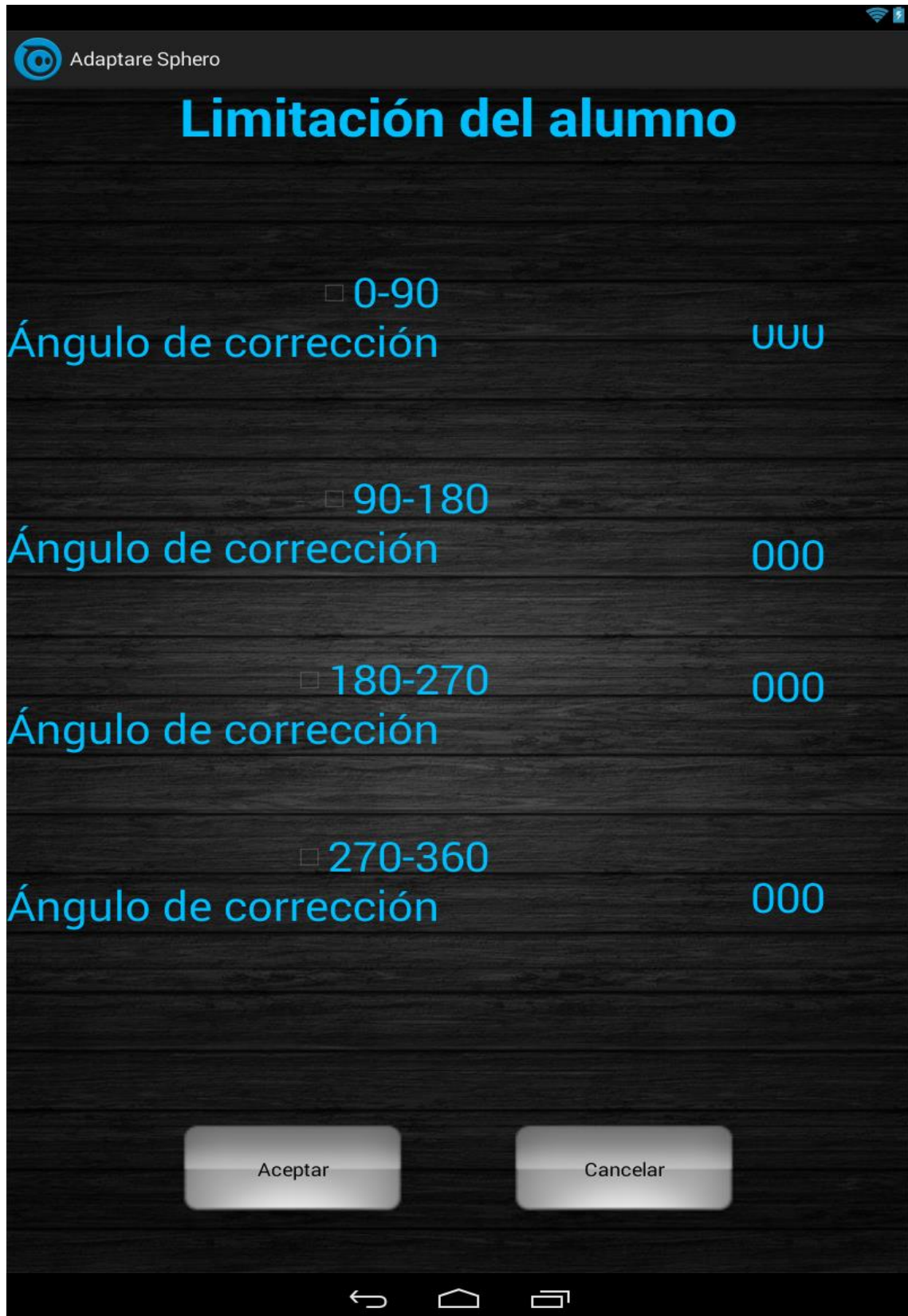


Figura 7.25: Pantalla correspondiente al Activity `LimitarActivity`

La clase corresponde a la “Activity LimitarActivity”, a la cual se accede tras haber pulsado el botón Limitar en la actividad anterior (Datos del Alumno). Consiste en una serie de movimientos que el profesor puede seleccionar como permitidos a realizar por el alumno, además de los ángulos de corrección de los mismos. Esta parte será utilizada en la actividad PosicionActivity, la cual será descrita más adelante.

Una de las ideas para este proyecto era poder limitar los movimientos que un alumno pudiera realizar mientras utilizaba la aplicación de control remoto del Sphero, añadiendo, además unos factores de corrección. Siguiendo esta necesidad, y plasmándolo en la pantalla de un dispositivo Android, se pensó en crear un plano cartesiano, quedando cuatro movimientos posibles. De esta forma, cuando un alumno marca hacia que lado mover el Sphero, si no está permitido el movimiento, no se enviará la instrucción correspondiente.

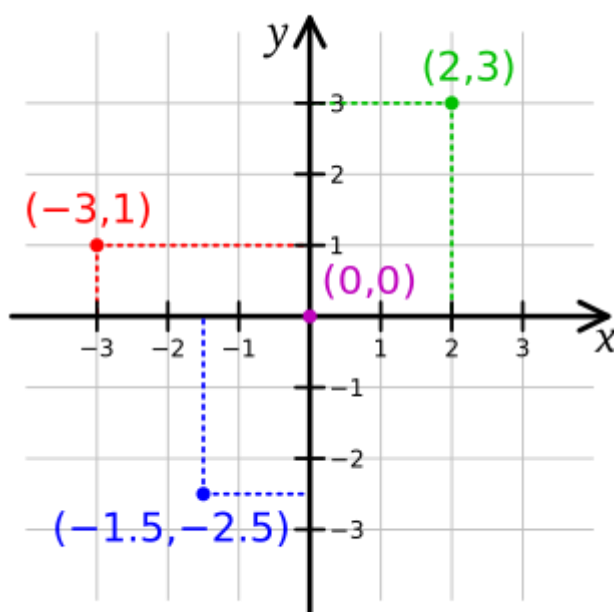


Figura 7.26: Plano cartesiano

Tal y como se puede observar en la imagen anterior, se pueden considerar cuatro posibles movimientos, suponiendo que el centro del espacio de deslizamiento en el dispositivo Tablet PC sea la coordenada (0,0), pudiendo desplazarse hacia los cuatro cuadrantes. Entonces,

si el movimiento ha sido marcado como viable en esta actividad, el Sphero recibirá la orden de movimiento enviada desde el dispositivo Android. De lo contrario, no recibirá nada ya que la aplicación no enviará ningún tipo de instrucción para comenzar el movimiento.

Completando esta funcionalidad, se pensó en posibles aplicaciones de esta técnica en el futuro. Una de ellas, en la que se basa el mando de la actividad PosicionActivity, es simular un joystick de movimiento de una silla de ruedas eléctrica. Esa palanca mueve la silla en función de la trayectoria hacia la que se va guiando. Pero existe un problema, y es que las personas con movilidad reducida puedan estar seleccionando de forma incorrecta la trayectoria, yendo, por ejemplo, hacia la derecha cuando quieren ir hacia delante.

Para solucionar esa problemática se pensó en poner factores de corrección, que son los grados de apertura desde el eje de coordenadas. Si, por ejemplo, una persona con movilidad reducida, en un alto porcentaje de veces intenta ir hacia adelante, pero el joystick no pasa de un ángulo de 30º, se puede poner factor de corrección inferior, de unos 15º, de forma que cada vez que se mueva el joystick con un ángulo inferior a ese factor, la silla se dirija hacia la derecha, y si es superior, avance hacia delante.

Para la parte visual se han utilizado un Relative Layout, cuatro Checkbox que marcados permiten los movimientos en esos cuadrantes, cuatro TextView para las etiquetas de los ángulos de corrección, cuatro EditText donde incluir los valores de los ángulos de corrección y dos botones en los que se carga una imagen y se asocia un método en el atributo onClick. Si se elige el botón Cancelar, se regresará a la actividad anterior sin actualizar los datos, y si se elige el botón Aceptar, se regresará a la actividad anterior actualizando los nuevos datos escritos.

Todos estos datos serán guardados en la memoria interna del teléfono, en un directorio determinado mediante un método. De esta forma, cada vez que se inicie la aplicación no será necesario volver a introducirlos.

7.2.2.8 Clase Cambio de contraseña

Accessibility Sphero

Cambiar Contraseña

Contraseña Actual:

Contraseña Nueva:

Repetir Contraseña: Nueva

Aceptar Cancelar

Figura 7.27: Pantalla correspondiente al Activity CambioPassword

La clase corresponde a la “Activity CambioPassword”, es a la cual se accede después de que el usuario haya pulsado el botón “Contraseña” en la actividad (ConfigAvanzada). Esta clase ha sido modificada con respecto a la primera versión, eliminando la parte relacionada con Token Flurry.

En la primera versión de la aplicación, se trabaja con Flurry Analytics para obtener las estadísticas de uso de la aplicación. Para esta segunda se decidió quitar este producto, ya que las estadísticas eran demasiado genéricas, y para poder aportar algo con ellas se realizó Adaptare. De esta forma, se eliminó de la actividad las TextView de Token Flurry, Flurry y el EditText donde se escribía el código de token asignado para este alumno.

El resto, toda la parte relacionada con el cambio de contraseña, se mantiene como en la primera versión. Para ello, aparecen los siguientes tres campos:

- Contraseña actual que exista en el sistema
- Nueva contraseña, a elegir
- Repetición de la nueva contraseña, para no validar una contraseña introducida por error.

Para esta actividad, se han utilizado un Relative Layout con el mismo fondo que la actividad anterior, cuatro Textview, para los diferentes campos relacionados con la modificación de la contraseña, tres EditText para la misma labor, y dos botones en los que se carga una imagen y se asocia un método en el atributo onClick. Si se elige el botón Cancelar, se regresará a la actividad anterior sin actualizar los datos, y si se elige el botón Aceptar, se regresará a la actividad anterior actualizando los nuevos datos escritos.

Todos estos datos serán guardados en la memoria interna del teléfono, en un directorio determinado mediante un método. De esta forma, cada vez que se inicie la aplicación no será necesario volver a introducirlos.

7.2.2.9 Clase Menú selección de mandos



Figura 7.28: Pantalla correspondiente al Activity MenuSelecMandos

La clase corresponde a la “Activity MenuSelecMandos”, es a la cual se accede después de que el usuario haya pulsado el icono de “Selección de Mando” en la actividad (MenuProfesor). Esta clase ha sido modificada para incluir los cinco nuevos mandos de los que dispone la aplicación en esta segunda versión.

Esta actividad hace uso de la clase “SlideHolder”, que implementa un menú swipe, en el que se establecen dos layouts, uno a la izquierda desplegable y otro a la derecha fijo.

El diseño de la parte visual se mantiene exactamente igual que en la primera versión, de forma que la colección de mandos aparece en el lado izquierdo, y en el derecho se decide si se añaden o no para el alumno. La funcionalidad tampoco ha cambiado, por lo que los cambios han sido mínimos con respecto a la primera versión, salvo una pequeña corrección comentada en el capítulo 8.

```
case R.id.toggleButtonCheck:

    if (toggleButtonCheck.getText().equals(
        getResources().getText(R.string.ToogleButtonAON))) {
        selectedControllers = selectedControllers + addController;
        checkActualButtonToggle(addController);
    }
    if (toggleButtonCheck.getText().equals(
        getResources().getText(R.string.ToogleButtonAOFF))) {
        if (selectedControllers.contains(addController)) {

            selectedControllers = selectedControllers.replace(
                addController, "");
        }
    }
    break;

public void checkActualButtonToggle(String addController) {
    if (addController.equals("Mando 1"))
        b1 = true;
    else
        b1 = false;
    if (addController.equals("Mando 2"))
        b2 = true;
    else
        b2 = false;
    if (addController.equals("Mando 3"))
        b3 = true;
    else
        b3 = false;
    if (addController.equals("Mando 4"))
        b4 = true;
    else
        b4 = false;
    if (addController.equals("Mando 5"))
        b5 = true;
    else
        b5 = false;
    if (addController.equals("Mando 6"))
        b6 = true;
```


7.2.2.10 Clase Lanzador de mando



Figura 7.29: Pantalla correspondiente al Activity LaunchController

La clase corresponde a la “Activity LaunchController”, es a la cual se accede después de que el usuario haya pulsado el icono de “Alumno” en la actividad (Main).

Mediante esta actividad, se producirá la sincronización con un Sphero, una vez que se haya sincronizado, va a obtener del Sphero un ID único, que posteriormente se utilizará para nuevas sincronizaciones cuando el usuario tenga que pasar de un mando a otro. Se almacenará en la memoria del dispositivo móvil, para recuperarla cuando interese.

```
public void onRobotConnected(Robot robot) {  
    mRobot = robot;  
    mSpheroConnectionView.setVisibility(View.GONE);  
    setUniqueID(mRobot.getUniqueId());  
    launchController();  
}
```

La primera modificación visible para el alumno en la aplicación consiste en el acceso a los ajustes de Android para activar el Bluetooth y sincronizarlo con el Sphero. En la primera versión de esta aplicación se tuvo en cuenta que, una vez seleccionado el modo alumno, si el Bluetooth estaba desactivado, se mostrase un mensaje de error informando de ello. Para esta segunda opción existían dos posibilidades:

- Activar el Bluetooth automáticamente, sincronizándolo directamente con el Sphero.
- Seguir el modelo de la aplicación oficial del fabricante Orbotix, mostrando el mensaje indicando que el Bluetooth está desactivado, mostrando una imagen con el símbolo del mismo, y, una vez pulsado, acceder a los ajustes de Android, de forma que queda en manos del usuario activarlo y sincronizarlo con el Sphero.

Después de analizar ambas opciones, nos decantamos por la segunda, ya que así depende del usuario la activación, siguiendo las recomendaciones de buenas prácticas.

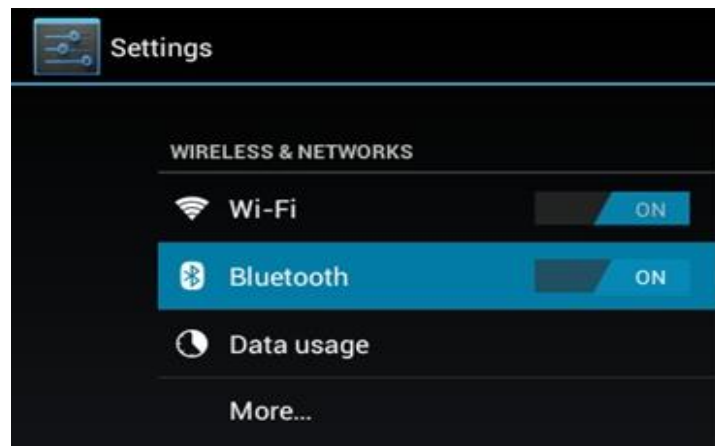


Figura 7.30: Activación de Bluetooth en dispositivo Android

Una vez que se activa el Bluetooth y se sincroniza con el Sphero, se puede regresar al punto de partida de la aplicación, de forma que el símbolo habrá desaparecido y ya se iniciará por completo.

```
@Override
public void onBluetoothNotEnabled() {

    Toast.makeText(LaunchController.this,
        "El Bluetooth no está habilitado",
        Toast.LENGTH_LONG).show();

    bBluetooth.setVisibility(View.VISIBLE);
}
```

7.2.2.11 Clase Mando joystick

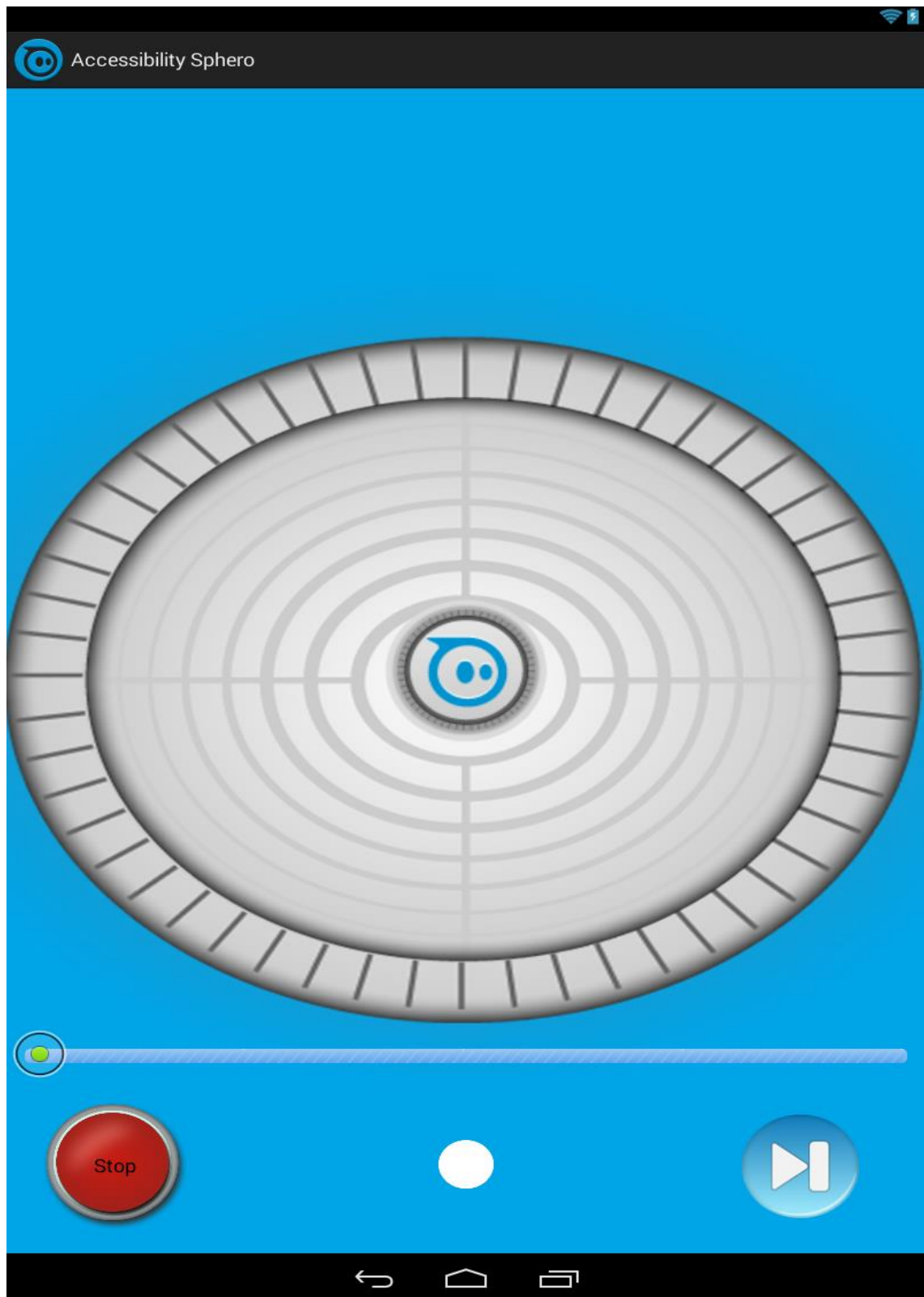


Figura 7.31: Pantalla correspondiente al Activity MJoystick

La clase corresponde a la “Activity MJoystick”, corresponde al primer mando de la lista, que ha sido seleccionado por el profesor en la actividad “MenuSelecMandos” y que ha sido lanzado desde la actividad “LaunchController”. Esta actividad no ha sido modificada, más allá de adaptar las funciones de navegación entre actividades, y del cambio de llamada al sistema de almacenamiento de estadísticas de uso Flurry por el nuevo desarrollado Adaptare.

Las características de este mando son las siguientes:

- Una zona circular, en la parte superior, que representa, la región por la que el joystick será desplazado.
- El joystick, representado por el icono e insignia de Sphero.
- Un control de velocidad, en el que es posible seleccionar, desde el 0% al 100%, dentro de un margen de velocidad ya establecido por el profesor en la actividad “ConfigAvanzada”.
- Un botón de parada, el cual, tras ser pulsado, envía a Sphero un mensaje indicando que debe detenerse inmediatamente. Al pulsarse, dicho botón cambia de estado y aparece con el aspecto de que ha sido pulsado.
- Un botón que permite calibrar la dirección de un Sphero. La función está contenida en el API de Sphero y simplemente invocando al método y añadiendo ese elemento el layout es posible utilizarlo. Permite rotar un Sphero sobre sí mismo para situar la cola en la posición deseada. Eso permite, tener un punto de referencia, a partir del cual la Sphero se desplazará siguiendo un ángulo de dirección.
- Un botón con una flecha hacia la derecha, que posibilita, el cambio al siguiente mando de la lista, seleccionado por el profesor. Si no existe ninguno más guardado, no hay posibilidad de cambiar de mando.

- Un menú de opciones, que contiene las opciones de mostrar el menú de ayuda, o lanzar el siguiente mando.

Es importante destacar, que se ha implementado de tres maneras diferentes el cambio al siguiente mando:

1. Mediante el botón descrito, que aparece en la pantalla con una flecha hacia la derecha.
2. Mediante el botón “Subir volumen” del lateral del dispositivo móvil.
3. Mediante el icono, de la parte superior derecha de la pantalla, que desplegará el Menú Bar y que, contendrá entre otras opciones, la opción de “Siguiente Mando”.

Si se parte de la aplicación oficial de Orbotix, este mando trata de simular un joystick similar al que propone el fabricante. De esta forma, siguiendo las coordenadas (X,Y), se irá calculando el ángulo hacia el que avanzar, de forma que el Sphero estará controlado en todo momento desde este mando.

```
relativeLayout.setOnTouchListener(new View.OnTouchListener() {  
  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
  
        valueOfX = event.getX();  
        valueOfY = event.getY();  
  
        heading = getHeadingValue(valueOfX, valueOfY);  
  
        joystick.setAdjustViewBounds(true);  
        joystick.setX(valueOfX);  
        joystick.setY(valueOfY);  
  
        // Roll robot  
        RollCommand.sendCommand(mRobot, heading, allowedVelocity);  
        FlurryAgent.logEvent("Ha usado el Joystick en dirección: "  
            + heading, userParams);  
        return true;  
    }  
});
```

Para enviar los movimientos al Sphero se utiliza el método `sendCommand` de la clase `RollCommand`. Este método tiene tres parámetros:

- **mRobot**: identificador del robot, mantenido durante la transición entre mandos.
- **heading**: dirección hacia la que girará el Sphero.
- **allowedVelocity**: velocidad de giro.

Estos tres parámetros son utilizados de forma muy dispar en los nuevos desarrollos. He aquí una reseña de los cambios comunes:

- **mRobot**: es el identificador del robot. Cada vez que se cambia de un mando a otro debe mantenerse para garantizar la sincronización de la aplicación con el juguete.

```
if (robot_id != null && !robot_id.equals("")) {  
    RobotProvider provider = RobotProvider.getDefaultProvider();  
    mRobot = provider.findRobot(robot_id);  
    provider.initiateConnection(robot_id);  
    provider.control(mRobot);  
    provider.connectControlledRobots();  
} else {  
    Toast.makeText(MJoystick.this, "Ningún Sphero sincronizado",  
        Toast.LENGTH_LONG).show();  
}
```

- **heading**: este parámetro se calcula de forma diferente para cada mando. Para los nuevos desarrollos se ha continuado con esa forma de trabajar, ya que en función de las necesidades el parámetro debe ser calculado de una forma u otra. Por ejemplo, no es lo mismo calcular los ángulos rectos en la actividad de `PosicionActivity` que los ángulos a obtener cuando se utilizan el juego de palabras con el que maneja la actividad de `VoiceActivity`.
- **allowedVelocity**: uno de los inconvenientes que tiene la primera versión de la aplicación es el difícil manejo del robot debido a la alta velocidad que alcanza. Si a esto le añadimos la dificultad que puede entrañar para una persona discapacitada, corremos el riesgo de

tener un juguete inmanejable. Ciertamente es que están desarrollados controladores de velocidad, pero la mínima obtenida siempre es demasiado elevada. Para evitar estas circunstancias, en los nuevos mandos se utilizan velocidades mucho más bajas, de forma que el manejo del Sphero es infinitamente más sencillo.

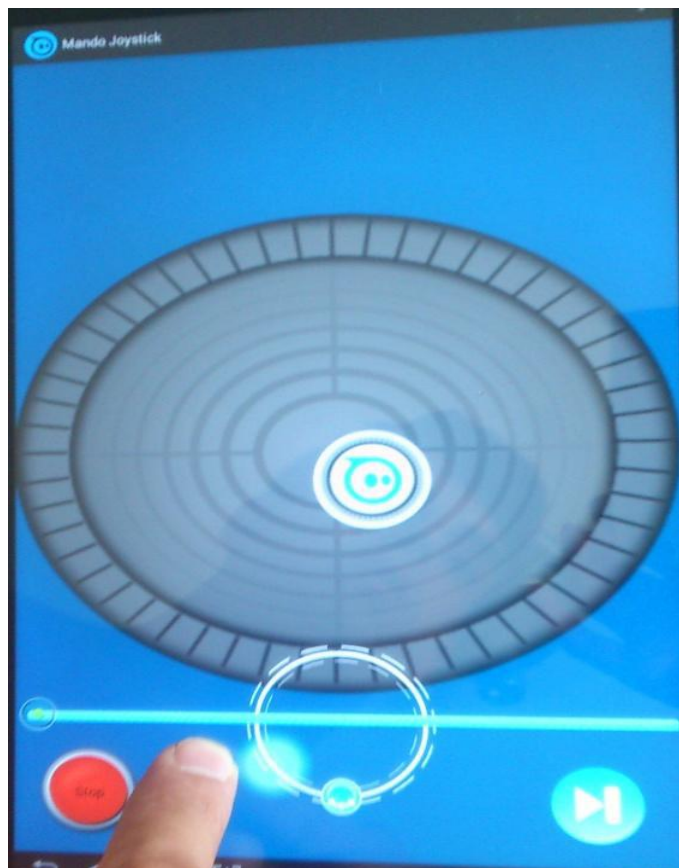


Figura 7.32: Captura de pantalla del uso del Calibrador de dirección

7.2.2.12 Clase Mando figuras predeterminadas



Figura 7.33: Pantalla correspondiente al Activity MFiguras

La clase corresponde a la “Activity Mfiguras”, corresponde al segundo mando de la lista, que ha sido seleccionado por el profesor en la actividad “MenuSelecMandos”, y puede ser accedida a través del botón de “Siguiete Mando” en caso de estar activo en la actividad anterior, o directamente al inicio de la aplicación. Esta actividad no ha sido modificada, más allá de adaptar las funciones de navegación entre actividades y del cambio de Flurry por Adaptare como sistema de almacenamiento de estadísticas de uso.

Con intención de facilitar la interacción con el juguete, en la primera versión se crearon una serie de movimientos ya definidos, de forma que el alumno solo tiene que elegir uno de los dibujos que aparecen en la pantalla de su Tablet PC para hacer mover el juguete Sphero. Existen diez reacciones diferentes en el Sphero, definidas a partir de cada uno de los siguientes botones:

- Un botón con una figura pre configurada, que al activarse, hace que el Sphero dibuje una trayectoria en forma de cuadrado, en la superficie en la que se encuentre.
- Un botón que al ser activado, produce una vibración en el Sphero y proyecta en él una luz roja.
- Un botón que al ser activado, hace que el Sphero realice un movimiento rotatorio sin desplazarse en la superficie.
- Un botón con una figura pre configurada, que al activarse, hace que el Sphero dibuje una trayectoria en forma de ocho, en la superficie en la que se encuentre.
- Un botón que al ser activado, hace que el Sphero realice un movimiento al que se ha identificado como “tornado”, porque realiza movimientos muy bruscos, con poco desplazamiento pero a gran velocidad y rotando sobre sí mismo.
- Un botón que al ser activado, hace que el Sphero cambie de color de forma progresiva.
- Un botón con una figura pre configurada, que al activarse, hace que el Sphero dibuje una trayectoria en forma de elipses entrelazadas.

- Un botón de parada, que una vez pulsado, hace que el Sphero se pare si está realizando cualquier función de los otros botones.
- Un botón con una flecha hacia la derecha, que posibilita, el cambio al siguiente mando de la lista, seleccionado por el profesor. Si no existe ninguno más guardado, no hay posibilidad de cambiar de mando.
- Un menú de opciones, que contiene las opciones de mostrar el menú de ayuda, o lanzar el siguiente mando.

7.2.2.13 Clase Mando porciones

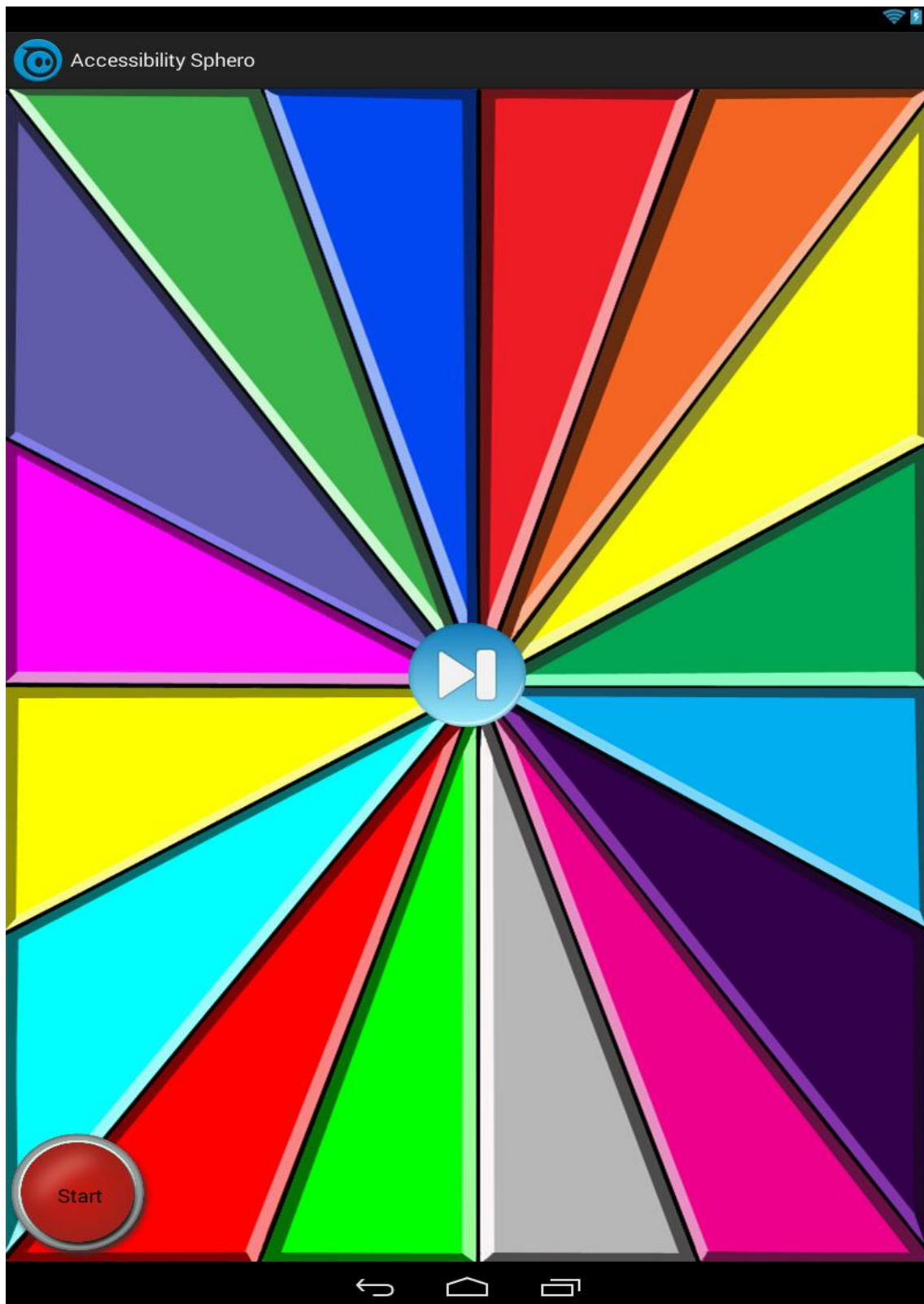


Figura 7.34: Pantalla correspondiente al Activity MPorciones

El tercer mando de la lista disponible para el profesor es este mando de Porciones. El profesor ha de seleccionarlo previamente para que aparezca durante la navegación entre mandos. La elección se realiza en la actividad “MenuSelecMandos”, y puede haber sido lanzada desde un “LaunchController” o desde un mando anterior, pulsando el botón de “Siguiende Mando”.

Este mando fue realizado para la primera versión de la aplicación para el control remoto del Sphero, pero su funcionalidad no tenía mucho que destacar. Simplemente, cuando un usuario pulsaba una de las porciones, el Sphero seguía esa dirección, siguiendo lo programado en las funciones de dirección del juguete.

El nuevo mando posee una nueva funcionalidad, aprovechando la serie de colores en los que puede iluminarse Sphero. Realmente se han creado dos funcionalidades, basadas en un antiguo juego de niños, llamado Simón:



Figura 7.35: Imagen del juego Simón

De esta forma, tenemos estas dos opciones:

- El juguete Sphero se ilumina del color que ha sido pulsado previamente, teniendo 16 posibilidades a elegir.
- El juguete Sphero repite la serie de colores que han sido marcados previamente, una vez pulsado el botón Inicio que aparece en la pantalla.

Aprovechando este cambio, una persona que esté ayudando a personas con discapacidad psíquica, podrá realizar actividades relacionadas con los colores, proponiendo series, y el usuario, podrá ver como el Sphero responde con las decisiones que él va tomando acerca de que color elegir.

Dicho mando contiene:

- Dieciséis botones que marcan un color a iluminar en el Sphero. Cada botón tiene un color diferente, que será el mismo que obtenga el Sphero cuando sea pulsado esa porción.
- Un botón con una flecha hacia la derecha, que posibilita, el cambio al siguiente mando de la lista, seleccionado por el profesor. Si no existe ninguno más guardado, no hay posibilidad de cambiar de mando.
- Un botón de inicio. Una vez pulsado este botón, el Sphero repetirá la secuencia de los colores desde la última vez que fue pulsado este botón.
- Un menú de opciones, que contiene las opciones de mostrar el menú de ayuda, o lanzar el siguiente mando.

Dentro de la parte visual, este mando no ha sido cambiado, salvo la desaparición de los botones de parada, el calibrador de dirección y del control de velocidad del Sphero. Teniendo en cuenta que el juguete robótico no se mueve en este mando, se ha considerado como lo mejor eliminar lo relacionado con funcionalidades antiguas. También, aprovechando el cambio, se ha eliminado el sistema de aviso de colisión, siguiendo la misma idea que para los botones anteriores.

Con respecto a este mando, los datos a enviar a Adaptare serán los siguientes:

- Usuario de la aplicación
- Color seleccionado
- Identificador del mando utilizado

Para poder enviar el color al Sphero, se siguen los siguientes pasos:

- Se aprovecha el método `getBitmapFromView` de la versión anterior, el cual obtiene del bitmap, gracias a que se puede conseguir en forma `RGBColor`.
- Con estos datos, se envía la combinación de los tres básicos al Sphero, simulando la porción elegida. Antes de enviar la información, se almacena la secuencia, para cuando se desee pulsar el botón de inicio.

```
aRedValue.add(rv.toString());  
aBlueValue.add(bv.toString());  
aGreenValue.add(gv.toString());  
  
// Set the color  
RGBLEDOutputCommand.sendCommand(mRobot, redValue, greenValue,  
    blueValue);
```

- Tanto para la selección única, como para la secuencia de colores, debe programarse unos segundos para poder observar el color, o de lo contrario, no daría tiempo suficiente a interpretarlos debidamente.

```
//Obtengo la combinación guardada
redValue2 = new Integer(aRedValue.get(i));
greenValue2 = new Integer(aGreenValue.get(i));
blueValue2 = new Integer(aBlueValue.get(i));

shapeMacro.addCommand(new RGB(redValue2, greenValue2, blueValue2, 0));
shapeMacro.addCommand(new Delay(delayValue));
```


7.2.2.14 Clase Mando paleta de colores

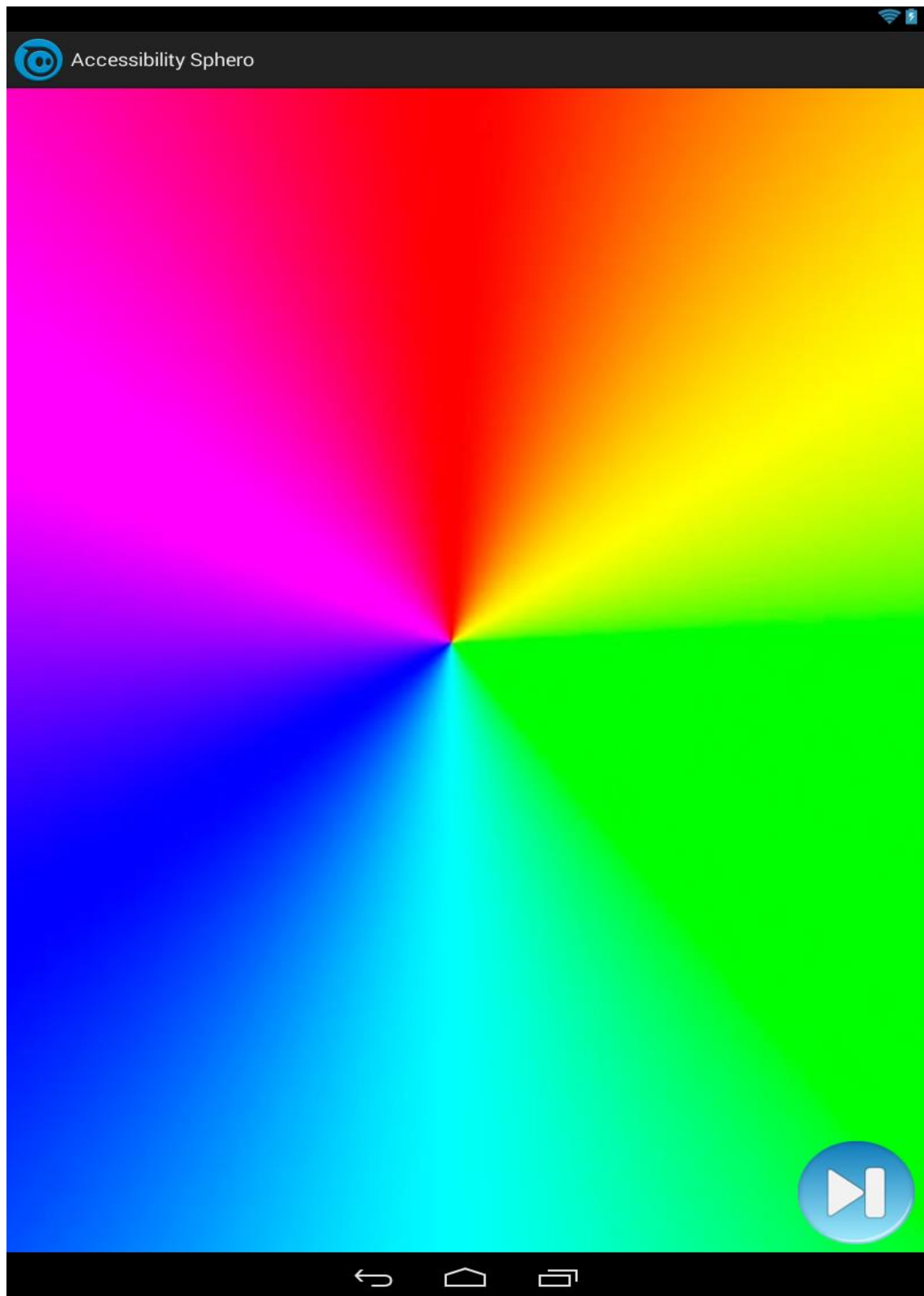


Figura 7.34: Pantalla correspondiente al Activity MPaleta

La clase corresponde a la “Activity MPaleta”, corresponde al cuarto mando de la lista, que ha sido seleccionado por el profesor en la actividad “MenuSelecMandos” y que puede haber sido lanzado desde la actividad “LaunchController” o desde un mando anterior, mediante el botón “Siguiente Mando”. Esta clase no ha sido modificada, más allá de modificar los métodos relacionados con la navegación entre mandos y el cambio de Flurry por Adapate como sistema de almacenamiento de datos de utilización.

En esta pantalla el usuario encontrará los siguientes elementos:

- Un layout con una imagen que presenta un amplio abanico de tonalidades.
- Un botón con una flecha hacia la derecha, que posibilita, el cambio al siguiente mando de la lista, seleccionado por el profesor. Si no existe ninguno más guardado, no hay posibilidad de cambiar de mando.
- Un menú de opciones, que contiene las opciones de mostrar el menú de ayuda, o lanzar el siguiente mando.

La principal característica de este mando es que se ha utilizado la propiedad del layout que permite utilizarlo como listener, de forma que en cada momento se va detectando que parte del mando se ha tocado, y a partir de ahí, obtener el color correspondiente al píxel tocado. Ese color es enviado al Sphero que, gracias a sus 16 millones de colores que puede llegar a proyectar, mostrará el color elegido.

```
relativeLayout.setOnTouchListener(new View.OnTouchListener() {  
  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
  
        int x = (int) event.getX();  
        int y = (int) event.getY();  
        int pixel = bitmap.getPixel(x, y);  
  
        int redValue = Color.red(pixel);  
        int blueValue = Color.blue(pixel);  
        int greenValue = Color.green(pixel);  
        // Set the color  
        RGBLEDOutputCommand.sendCommand(mRobot, redValue, greenValue,  
                                         blueValue);  
  
        return true;  
    }  
});
```

7.2.2.15 Clase Mando por voz

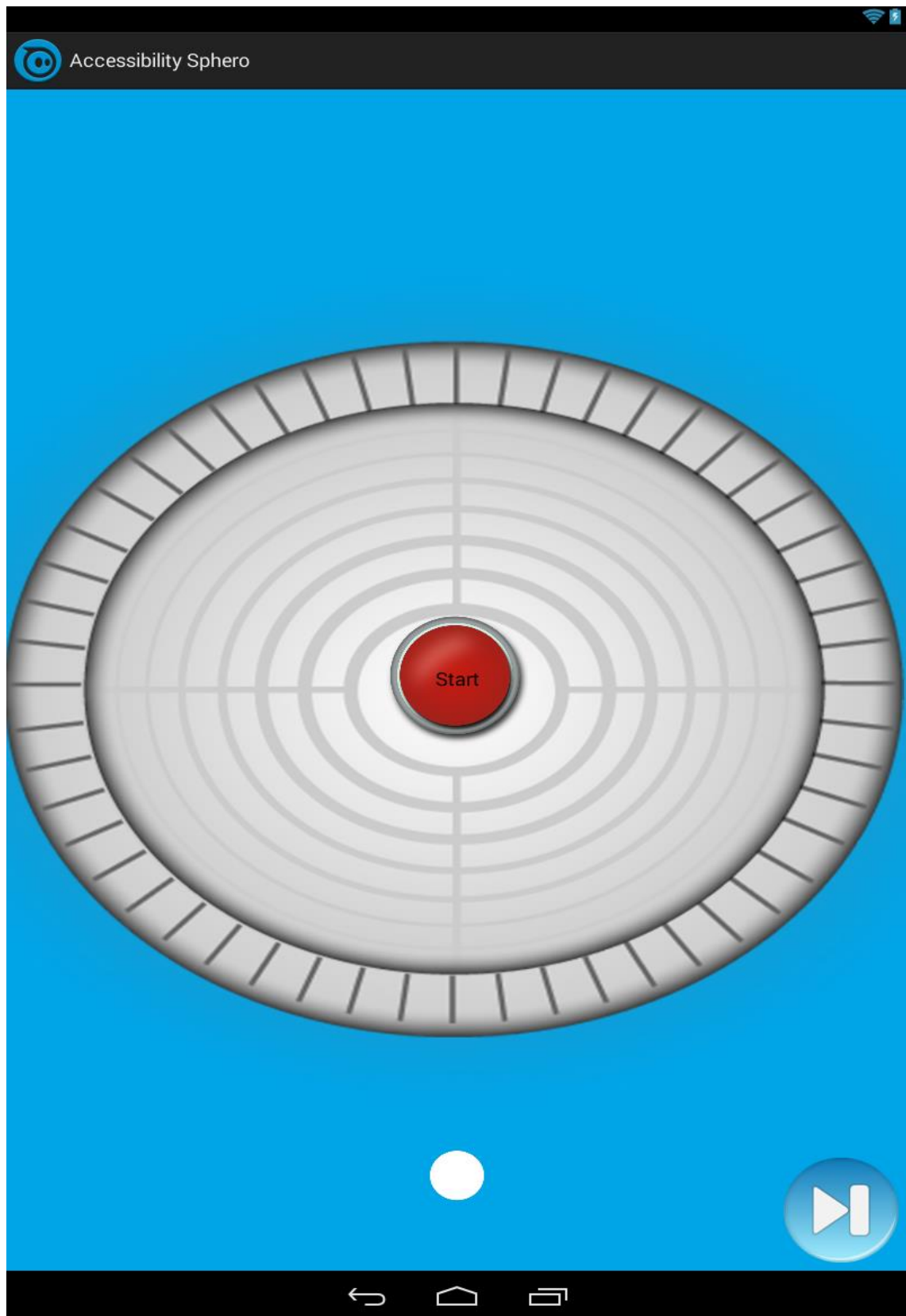


Figura 7.36: Pantalla correspondiente al Activity VoiceActivity

Uno de los nuevos mandos que aparecían como requisitos de esta nueva versión consistía en realizar un mando que recibiera instrucciones por voz y fueran ejecutadas por el Sphero. Esta clase, que corresponde a la Activity VoiceActivity, es la respuesta a esa necesidad.

Para la realización del mismo se ha utilizado el paquete de Android llamado android.speech. Aprovechando las ventajas que este paquete ofrece, el usuario, tras pulsar el botón de activación de reconocimiento de voz, podrá indicar una instrucción que el Sphero ejecutará. En el caso de no reconocer el concepto, o, no estar entre las instrucciones definidas, el Sphero no realizará ninguna acción.

Otra de las características del funcionamiento es que la instrucción debe comunicarse de forma clara. Cuando android.speech devuelve los resultados de las palabras asociadas al sonido reconocido, lo hace en forma de listado de resultados. Para esta versión, solo se tiene en cuenta el primero de la lista, ya que por defecto, es el más parecido a lo emitido por el usuario.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_CODE && resultCode == RESULT_OK) {
        ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

        String lectura= new String(matches.get(0).toString().toLowerCase());
```

El funcionamiento es el siguiente: el usuario debe pulsar el botón de activación de reconocimiento de voz, tiene unos segundos para poder enviar un sonido al dispositivo Android. Si no lo hace, se terminará el plazo dispuesto para ello, teniendo que pulsar de nuevo el botón para comenzar el proceso. De hacerlo, se obtendrán una serie de resultados, analizando en el código solo el primero de ellos. Si no coincide con los definidos, aparecerá el mensaje de error indicando este hecho. De lo contrario, se enviará la instrucción al Sphero, que este ejecutará:

```
if(lectura != null) {  
    if (PARA.equals(lectura)) {  
        stopMacroVoice();  
    } else if (STOP.equals(lectura)) {  
        stopMacroVoice();  
    } else if ((getHeadingWord(lectura) != 1) && (getHeadingWord(lectura) < 1000) ) {  
        stopMacroVoice();  
        if(getHeadingWord(lectura)== 2) {  
            spinMacroClicked();  
        } else if (getHeadingWord(lectura)== 3) {  
            flipMacroClicked();  
        } else if (getHeadingWord(lectura)== 4) {  
            vibrateMacroClicked();  
        } else {  
            heading = getHeadingWord(lectura);  
            RollCommand.sendCommand(mRobot, heading, 0.3f);  
        }  
    } else if ((getHeadingWord(lectura) != 1) && (getHeadingWord(lectura) >= 1000) ) {
```

Por otro lado, completando las instrucciones de movimiento, y aprovechando los recursos que ofrece el juguete Sphero, también se ha decidido hacer un listado de colores, los cuales, a través de a instrucción con su nombre, se enviarán al Sphero y este adquirirá el color indicado:

```
} else if (getHeadingWord(lectura) == 1004) {  
    //MARRON  
    redValue = 102;  
    greenValue = 51;  
    blueValue = 0;  
    // Set the color  
    RGBLEOutputCommand.sendCommand(mRobot, redValue, greenValue,  
        blueValue);  
} else if (getHeadingWord(lectura) == 1005) {  
    //NARANJA  
    redValue = 255;  
    greenValue = 128;  
    blueValue = 0;  
    // Set the color  
    RGBLEOutputCommand.sendCommand(mRobot, redValue, greenValue,  
        blueValue);  
} else if (getHeadingWord(lectura) == 1006) {  
    //AMARILLO  
    redValue = 255;  
    greenValue = 255;  
    blueValue = 0;  
    // Set the color  
    RGBLEOutputCommand.sendCommand(mRobot, redValue, greenValue,
```

Para esta versión, se han definido esta serie de instrucciones:

- **Adelante:** el Sphero rodará hacia adelante, siendo marcada la trayectoria por la luz de cola.
- **Atrás:** el Sphero rodará en el sentido opuesto que en la instrucción anterior.
- **Derecha:** el juguete rodará hacia el lado derecho.
- **Izquierda:** el juguete rodará en el sentido opuesto de la instrucción anterior.
- **Para:** el Sphero se detendrá.
- **Stop:** el Sphero se detendrá.
- **Gira:** el Sphero repetirá uno de los movimientos definidos en el mando de figuras.
- **Voltea:** el Sphero repetirá uno de los movimientos definidos en el mando de figuras.
- **Vibra:** el Sphero vibrará, siguiendo el movimiento definido en el mando de figuras.
- **Rojo:** el Sphero se iluminará de color rojo.
- **Azul:** el Sphero se iluminará de color azul.
- **Verde:** el Sphero se iluminará de color verde.
- **Negro:** el Sphero se iluminará de color negro.
- **Marrón:** el Sphero se iluminará de color marrón.

- **Naranja:** el Sphero se iluminará de color naranja.
- **Amarillo:** el Sphero se iluminará de color amarillo.
- **Rosa:** el Sphero se iluminará de color rosa.
- **Morado:** el Sphero se iluminará de color morado.
- **Norte:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Sur:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Este:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Oeste:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Noroeste:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Nordeste:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Suroeste:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.
- **Sureste:** el Sphero se moverá en la dirección siguiendo el punto cardinal indicado.

Las instrucciones indicadas mediante puntos cardinales fueron incluidas porque podrían facilitar el manejo del Sphero para personas con discapacidad. Siguiendo su ubicación en un dibujo, pueden asociar los movimientos a la imagen que están mirando, de forma que tienen ubicado al juguete continuamente:



Figura 7.37: Puntos cardinales

Es importante señalar el hecho del botón de luz de cola del Sphero. Tal y como se indica en la información del juguete, la dirección de movimiento será marcada por la luz de cola, mientras que el sentido será el contrario al marcado por misma. De esta forma, la palabra “Adelante” o “Norte” harán que el juguete comience a rodar en el sentido contrario al indicado por la luz de cola. El resto de movimientos, se interpretarán siguiendo esta instrucción, completándola con lo descrito más arriba.

```
mCalibrationView = (CalibrationView) findViewById(R.id.calibration_view);  
mCalibrationImageButtonView = (CalibrationImageButtonView) findViewById(R.id.calibration_image_button);  
mCalibrationImageButtonView.setCalibrationView(mCalibrationView);  
mCalibrationImageButtonView.setRadius(100);  
mCalibrationImageButtonView  
    .setOrientation(CalibrationView.CalibrationCircleLocation.ABOVE);
```

También, aprovechando el mando de figuras, con algunos movimientos definidos, se han aprovechado tres de ellos para esta parte, asignados a las palabras escritas más arriba:

```
public void flipMacroClicked() {  
    // returnSpheroToStableState();  
  
    // Create a new macro object to send to Sphero  
    MacroObject flipMacro = new MacroObject();  
    flipMacro.addCommand(new RGB(0, 0, 255, 0));  
    // You must turn stabilization off to use the raw motors  
    flipMacro.addCommand(new Stabilization(false, 0));  
    // Run both motors forward at full power  
    flipMacro.addCommand(new RawMotor(RawMotor.DriveMode.FORWARD, 255,  
        RawMotor.DriveMode.FORWARD, 255, 0));  
    // Delay for a certain time period  
    flipMacro.addCommand(new Delay(delayValue));  
    // Remember to turn stabilization back on  
    flipMacro.addCommand(new Stabilization(true, 0));  
    flipMacro.addCommand(new RGB(0, 255, 0, 0));  
    flipMacro.setMode(MacroObject.MacroObjectMode.Normal);  
    //FlurryAgent.logEvent("Usa el botón Rotación", userParams);  
    flipMacro.setRobot(mRobot);  
    flipMacro.playMacro();  
}
```

Figura 7.38: Código para la instrucción Voltea

Con respecto a la parte visual, se ha utilizado un Relative Layout con un botón en el que se carga una imagen, un TextView para indicar que se debe pulsar el botón para comenzar a recibir las instrucciones y el mismo fondo que el mando que simula un joystick, por mantener el mismo estilo visual que el primer mando de la primera versión. Por último, tal y como ocurre en todos los mandos, los datos de las instrucciones ejecutadas serán enviadas al sistema Adaptare para su almacenamiento y posterior consulta.

7.2.2.16 Clase Mando por movimiento del dispositivo Android

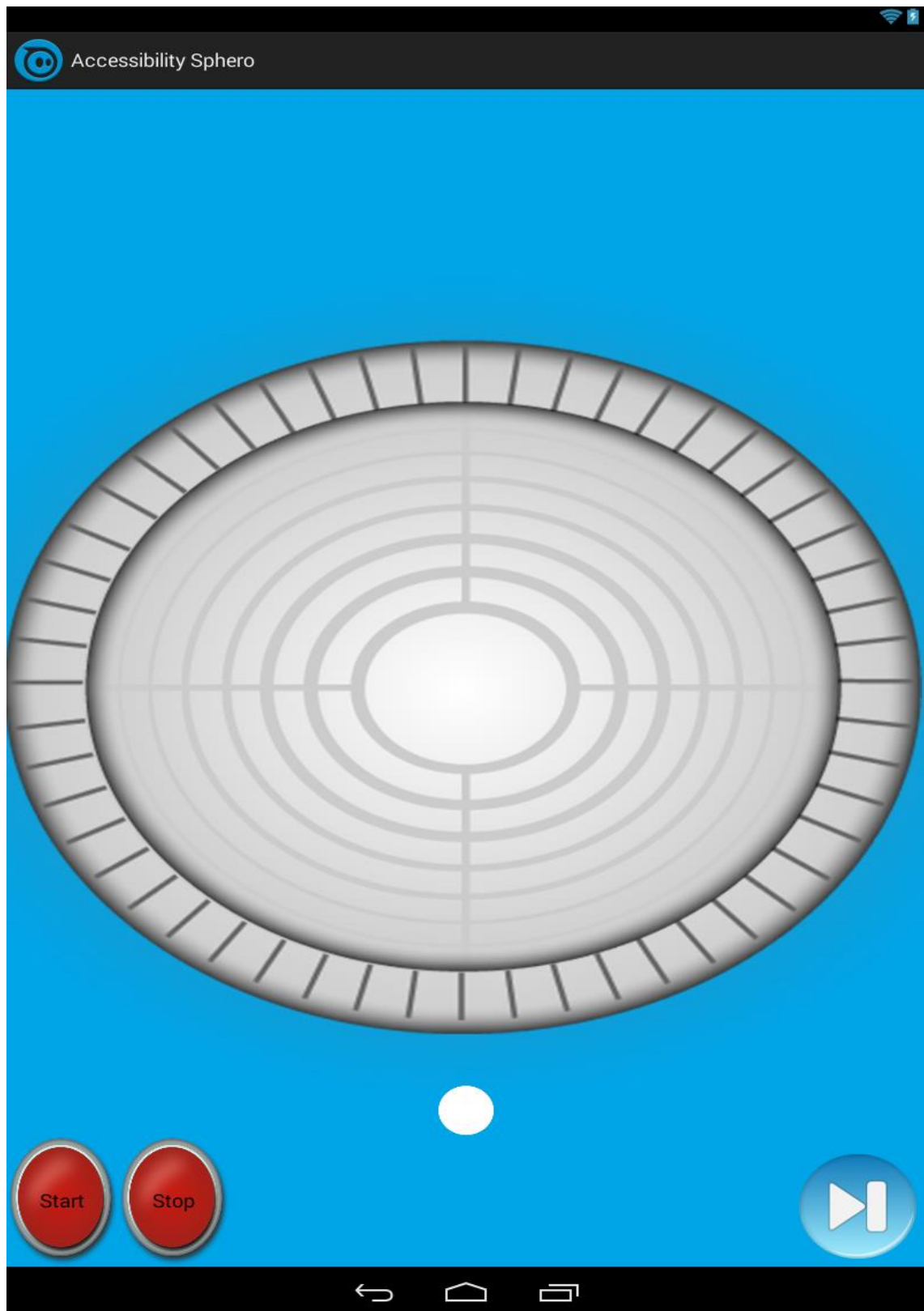


Figura 7.39: Pantalla correspondiente al Activity SensorActivity

Otro de los nuevos mandos a realizar consistía en ver la posibilidad de aprovechar el acelerómetro del dispositivo Android para, siguiendo sus movimientos, dirigir el juguete como si fuera un joystick. La clase correspondiente a la “Activity SensorActivity” responde a esta necesidad.

Cuando una persona con movilidad reducida tiene intención de utilizar la aplicación para mover el juguete Sphero se puede encontrar con la dificultad que le entraña mover el joystick del primer mando realizado, más aún en dispositivos Android de menor tamaño, como son algunos teléfonos inteligentes. Para solucionar este problema, y dar cabida a un mayor número de personas que puedan utilizar esta aplicación, se pensó que podría tener un uso más sencillo el mover todo el dispositivo y que este enviara las instrucciones correspondientes al juguete. Aprovechando el acelerómetro que poseen los dispositivos Android, se dio solución a esta necesidad.

Siguiendo su definición técnica, un acelerómetro es un dispositivo que mide la aceleración y las fuerzas inducidas por la gravedad. Gracias a esto, permite detectar el movimiento y el giro. De esta forma, puede comprender en qué posición está el dispositivo Android. Uno de los ejemplos claros de su uso, es el cambio de posición del contenido de la pantalla, de vertical a horizontal y viceversa, dependiendo en que lado se esté sosteniendo el aparato.



Figura 7.40: Imagen de un acelerómetro

Su funcionamiento es el siguiente: posee un tubo con un muelle y una masa en su interior. La masa, al moverse el dispositivo, se desplaza de forma vertical, por lo que con la interpretación conjunta de los ejes se puede conocer la posición del móvil.

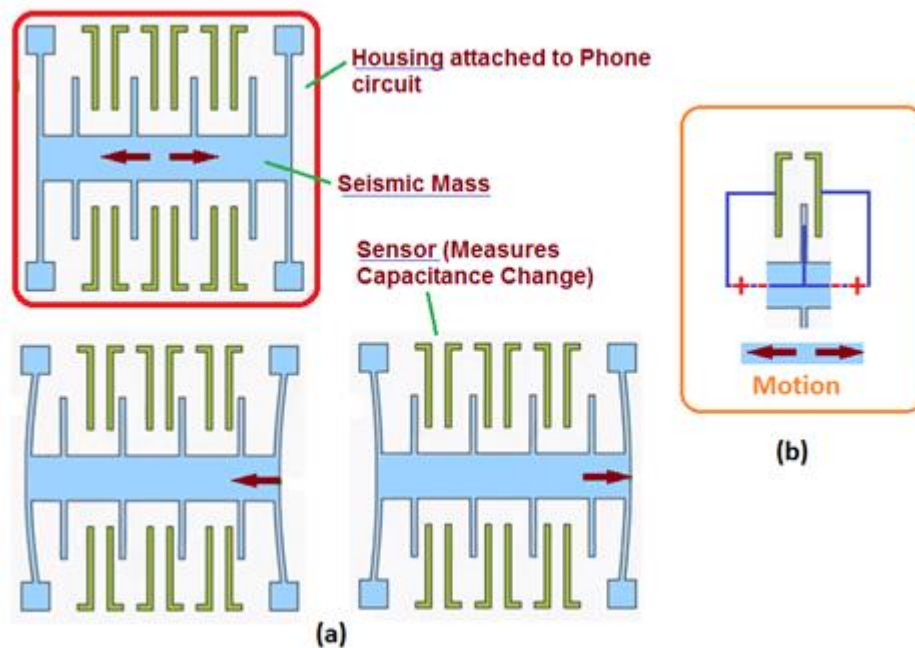


Figura 7.41: Descripción del funcionamiento de un acelerómetro

Existen dos tipos de acelerómetros: de dos y tres dimensiones. La diferencia, además de estar en su estructura, está en sus capacidades y funciones. El de tres dimensiones es el más robusto y más utilizado de ambos.

Además de sus posibles funcionalidades, las cuales potencia no solo las posibilidades del manejo del Sphero, sino todas las de un teléfono inteligente o Tablet PC, otra de sus partes positivas es el precio tan reducido que tiene, por debajo de un euro por cada uno de ellos.

Su inclusión en el proyecto es sencilla: se ha realizado un mando, el cual posee un botón de activación de envío de movimientos al Sphero, y otro para detenerlos. De esta forma, una vez accedido al mando, el Sphero no se volverá incontrolado moviéndose continuamente, posibilitando mover el dispositivo Android sin que el Sphero siga sus movimientos previamente al comienzo real del manejo:

```
protected void onResume()
{
    super.onResume();
    SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
    List<Sensor> sensors = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
    if (sensors.size() > 0) //dispositivo android tiene acelerometro
    {
        sm.registerListener(this, sensors.get(0), SensorManager.SENSOR_DELAY_GAME);
    }
}
```

Para facilitar el control del Sphero, solamente se han registrado ocho movimientos posibles, siguiendo la idea anterior de los puntos cardinales. De esta forma, en función de los datos de coordenadas entregadas por el acelerómetro a la aplicación, seguirá uno de los caminos descritos.

```
public void onSensorChanged(SensorEvent event) {
    valueOfX = event.values[SensorManager.DATA_X];
    valueOfY = event.values[SensorManager.DATA_Y];

    if(startClicked){
        heading = getHeadingValue(valueOfX, valueOfY);
        // Roll robot
        RollCommand.sendCommand(mRobot, heading, 0.3f);
    }
}
```

Otro de las cosas a tener en cuenta a la hora de diseñar este mando es el tipo de sensibilidad que debe tener. El acelerómetro está detectando cambios continuamente, pero si esos cambios no son bruscos, es decir, obtienen una serie de valores elevados, no serán aceptados como tal, por lo que no se enviarán al Sphero. De esta forma, una vez más, se facilita el control para personas de movilidad reducida, las cuales podrían llegar a tener problemas si la sensibilidad de los movimientos fuera enviada continuamente al Sphero.

```
protected float getHeadingValue (float x, float y) {  
    float head = 0;  
  
    //Adelante - Norte  
  
    if (x > 4.5 && y > -1.5 && y < 1.5)  
    {  
        //head = 270f; //180f Acer  
        head = 0f;  
        movimiento = "Norte";  
    }  
  
    //Atrás - Sur  
    if (x < -4.5 && y > -1.5 && y < 1.5)  
    {  
        //head = 90f; //0f Acer  
        head = 180f; //0f Acer  
        movimiento = "Sur";  
    }  
  
    //Derecha - Este  
    if (y < -4.5 && x > -1.5 && x < 1.5)  
    {  
        head = 90f; //90f Acer  
        movimiento = "Este";  
    }  
  
    //Izquierda - Oeste  
    if (y > 4.5 && x > -1.5 && x < 1.5)  
    {  
        head = 270f; //270 Acer  
        movimiento = "Oeste";  
    }  
}
```

Es importante señalar el hecho del botón de luz de cola del Sphero. Tal y como se indica en la información del juguete, la dirección de movimiento será marcada por la luz de cola, mientras que el sentido será el contrario al marcado por misma. De esta forma, al inclinar el dispositivo Android hacia delante, el juguete comenzará a rodar en el sentido contrario al indicado por la luz de cola. El resto de movimientos, se interpretarán siguiendo esta instrucción.

```
mCalibrationView = (CalibrationView) findViewById(R.id.calibration_view);  
mCalibrationImageButtonView = (CalibrationImageButtonView) findViewById(R.id.calibration_image_button);  
mCalibrationImageButtonView.setCalibrationView(mCalibrationView);  
mCalibrationImageButtonView.setRadius(100);  
mCalibrationImageButtonView  
    .setOrientation(CalibrationView.CalibrationCircleLocation.ABOVE);
```

En la parte visual simplemente se ha utilizado un Relative Layout, con los botones en los que se carga una imagen indicados anteriormente, y el botón de cambio de mando.

Al igual que en el resto de mandos de la aplicación, los datos de uso, tanto del mando utilizado como de los movimientos realizados, serán enviados al sistema Adaptare para el almacenamiento y la consulta de los mismos, siendo asociados al usuario de la aplicación.

7.2.2.17 Clase Mando por dibujo de trayectoria



Figura 7.42: Pantalla correspondiente al Activity TrayectoriaLayout

El tercer nuevo mando solicitado para este proyecto consistía en dibujar una trayectoria en el layout del dispositivo Android, y que el Sphero repitiese esa trayectoria dibujada. La clase correspondiente a la “Activity TrayectoriaLayout” es la que soporta esta funcionalidad.

Cuando se piensa en personas discapacitadas, ya sea de forma física o psíquica, una de las primeras cosas que se deben tener en cuenta es como poder facilitarle el día a día. En este caso, para algunas de ellas, el manejo del Joystick del primer mando de la aplicación puede llegar a ser complicado, de forma que no se obtendrá el objetivo de poder jugar con el Sphero. Para ello, además de los dos mandos anteriores, los cuales, facilitan en gran medida el control del juguete, se pensó que dibujando una trayectoria en el layout del dispositivo Android y enviando la misma al Sphero, el control podría ser más fácil. Además, puede utilizarse la imaginación para dibujar cualquier tipo de figura, e incluso, tratar de repetirse, de forma manual, algunos de los movimientos definidos en el mando de figuras.



Figura 7.43: Clase Canvas para desarrolladores Android

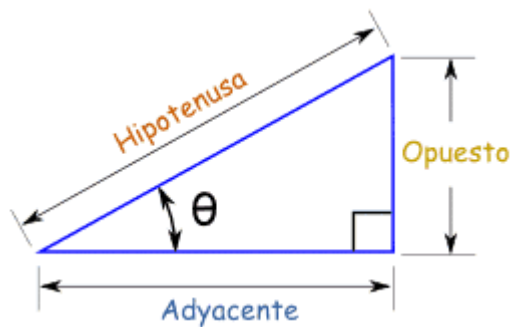
Para la realización de este mando se ha utilizado el paquete android.graphics, en concreto, su clase Canvas. De esta forma, se pueden ir recogiendo las coordenadas dibujadas en la pantalla, generar el ángulo de ataque del Sphero gracias a ellas y enviar el movimiento para que este lo repita. Al poderse realizar movimientos muy alargados en el tiempo, se ha considerado poner un tiempo máximo de rodadura del juguete robótico.

La clase Canvas (lienzo en inglés) permite a los desarrolladores representar una superficie donde se puede dibujar. Para cada uno de los dibujos posibles, líneas, círculos, etc, dispone de un método para cada uno de ellos. Esas figuras, como cualquier otro dibujo, debe realizarse con la ayuda de un pincel, a definir dentro del código en todas sus características.

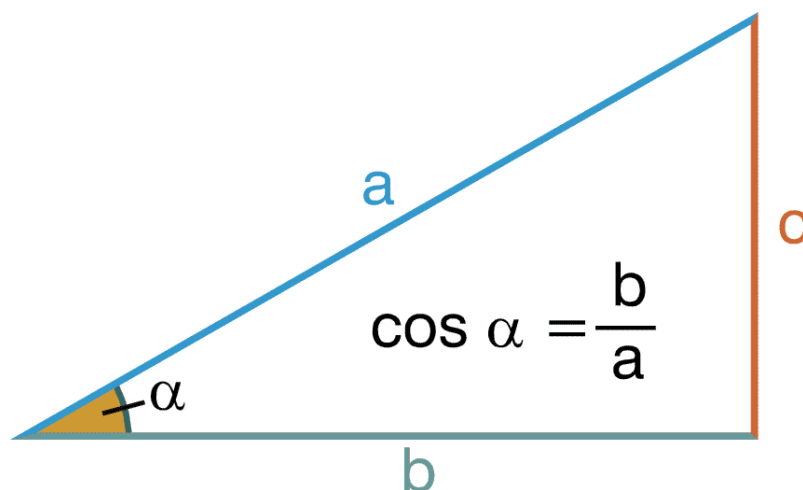
En la API de Canvas, se pueden ver todos los métodos anteriormente señalados, además de las características de cada uno de ellos.

A nivel funcional, este mando tiene varios problemas que han ido resolviéndose a medida que se desarrollaba la versión final. Estos son algunos de ellos, con la correspondiente solución:

- **Ángulo de orientación:** mientras que en otros mandos los métodos que calculaban el ángulo de orientación del Sphero no contemplaban todos los valores, si se desea realizar una figura completa, hay que tener en cuenta todos los puntos de la misma. Para poder orientar al Sphero de un punto al siguiente, debe calcularse el ángulo de orientación que debe adquirir. Para ello, se ha utilizado trigonometría, suponiendo que el centro del panel donde se dibuja la trayectoria es el centro de unos ejes de abscisas y ordenadas:



Dibujando un triángulo a partir de ese centro, y utilizando el avance de las coordenadas X e Y, se puede obtener el coseno de ese ángulo:



Al ser un triángulo rectángulo, para obtener el valor de la hipotenusa, necesario para calcular el coseno del ángulo de orientación, se ha utilizado el teorema de Pitágoras:

```
result = Math.pow(valueOfX - aX,2) + Math.pow(valueOfY - aY,2);  
resultT = Math.sqrt(result);
```

Una vez obtenido el coseno del ángulo, aplicando su función recíproca, el arcocoseno, se obtiene el ángulo de orientación:

```
if (valueOfY < aY) {  
    if (valueOfX < aX) {  
        anguloL = new Float(Math.toDegrees(Math.acos((aX - valueOfX) / resultT))) + 90;  
    } else if (valueOfX > aX) {  
        anguloL = new Float(Math.toDegrees(Math.acos((valueOfX - aX) / resultT)));  
    }  
} else {  
    if (valueOfX < aX) {  
        anguloL = new Float(Math.toDegrees(Math.acos((aX - valueOfX) / resultT))) + 180;  
    } else if (valueOfX > aX) {  
        anguloL = new Float(Math.toDegrees(Math.acos((valueOfX - aX) / resultT))) + 270;  
    }  
}
```

Como se puede ver en la imagen anterior, se aplica una suma de un número de grados, una vez convertidos de radianes, para situar al Sphero en el cuadrante adecuado y su movimiento sea lo más fiel posible al dibujo realizado en el layout:

- **Espacio de ejecución:** debido a que el juguete se puede utilizar en espacios pequeños cerrados, se ha tratado de hacer sus movimientos a escala, de forma que en cualquier habitación de una casa pueda ejecutarse la trayectoria completa con total normalidad. Al ser un layout dibujado en píxeles, si se sigue cada uno de ellos para el dibujo de la trayectoria, el número de ellos que recorre puede ser muy alto. Si tenemos en cuenta el espacio que necesita el juguete para realizar sus maniobras, se puede convertir en un juguete a utilizar solamente en grandes espacios abiertos, y esa no es la intención.



Para solucionar este problema, tras varias pruebas realizadas con diferentes factores, se llegó a la conclusión de que el tiempo determinado para cada uno de los movimientos a realizar entre dos puntos de la trayectoria debía ser de 1 segundo. La combinación de este tiempo, más el avance por cada uno de los puntos de la trayectoria dibujada en el layout tiene como resultado una trayectoria se asemeja casi en su totalidad a la dibujada en el layout. Su ejecución, de esta forma, es mucho más rápida, pudiendo completarse en un espacio bastante más reducido del necesitado en un principio.

- **Tiempo de ejecución:** además de reducir algunos puntos, como se explica en el punto anterior, otro factor importante era el tiempo de ejecución de la trayectoria por cada ángulo de orientación. Si se dibuja una trayectoria grande, además de los problemas físicos de espacio, su finalización podría retrasarse demasiado. Para ello, al estar trabajando en este mando con métodos Macro y no con métodos de ejecución sencillos, se ha decidido que el tiempo a ejecutarse cada uno de los movimientos de la macro ha de ser de 1000 milisegundos. De esta forma no se eterniza la ejecución de la trayectoria:

```
while (i < aValoresX.size()){  
    // Change Color  
    valueOfX = new Float(aValoresX.get(i));  
    valueOfY = new Float(aValoresY.get(i));  
    shapeMacro.addCommand(new RGB(0, 255, 0, 255));  
    // Change direction to roll in increments of a 360  
    heading2 = getHeadingValue(valueOfX, valueOfY);  
    shapeMacro.addCommand(new Roll(0.4f, heading2, delayValue));  
    aX = valueOfX;  
    aY = valueOfY;  
    i = i + 1; //12;  
}  
  
shapeMacro.addCommand(new Roll(0.0f, 0, 255));  
  
// Set Macro size  
shapeMacro.setMode(MacroObject.MacroObjectMode.Normal);  
//FlurryAgent.logEvent("Usa el botón Formas Predefinidas", userParams);  
shapeMacro.setRobot(mRobot);  
// Send Macro  
shapeMacro.playMacro();  
  
RollCommand.sendStop(mRobot);
```

En la imagen se puede apreciar un cambio de color en el Sphero durante la ejecución de la trayectoria. Esta decisión se ha tomado para llamar la atención de la persona que está utilizando el juguete, para que pueda prestar más interés en ver como se repite lo dibujado en el layout de la Tablet PC.

En la parte visual del mando, se puede ver un lienzo sobre el que se dibuja la trayectoria a seguir, y tres botones que cargan una imagen, dos para el manejo del Sphero (comenzar a rodar siguiendo la trayectoria, y, eliminar la trayectoria dibujada previamente) y uno para cambiar de mando.

7.2.2.18 Clase Mando de recepción de movimientos del Sphero

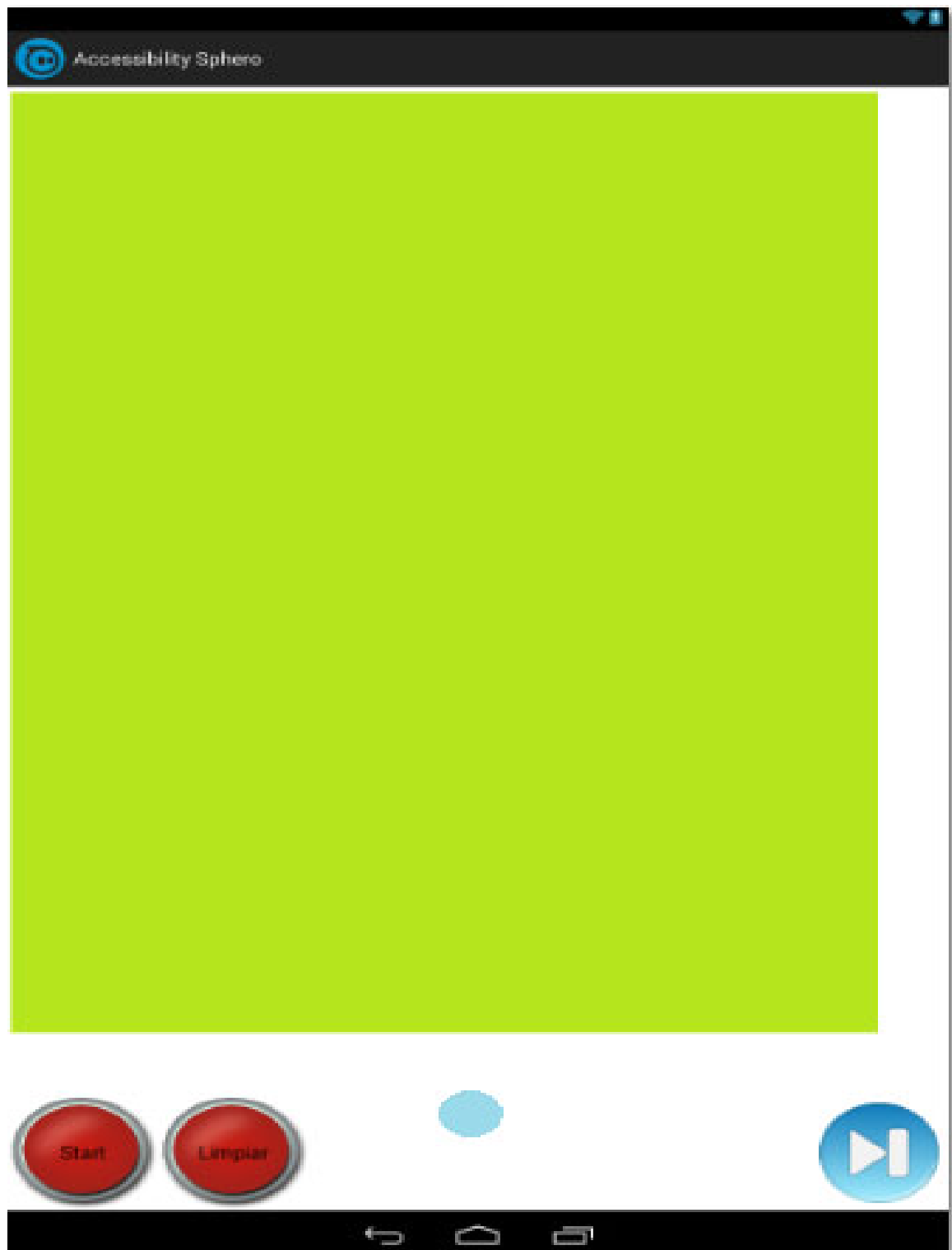


Figura 7.44: Pantalla correspondiente al Activity LocatorActivity

Basado en el mando anterior, el cuarto mando se pensó en realizar la función inversa, es decir, aprovechando las propiedades de Sphero, en concreto, la de su posición en cada momento, dibujar la trayectoria dibujada con el juguete a modo de ratón de PC en la pantalla del dispositivo Android. La clase aparecida corresponde a la “Activity LocatorActivity”.

Si recordamos las posibilidades de movimiento del Sphero, nos encontramos que tiene tres ejes sobre los que se mueve. Para este caso, al utilizar la clase LocatorData de la API de Orbotix para obtener la posición, solamente podemos obtener los valores de la posición en las coordenadas X e Y.

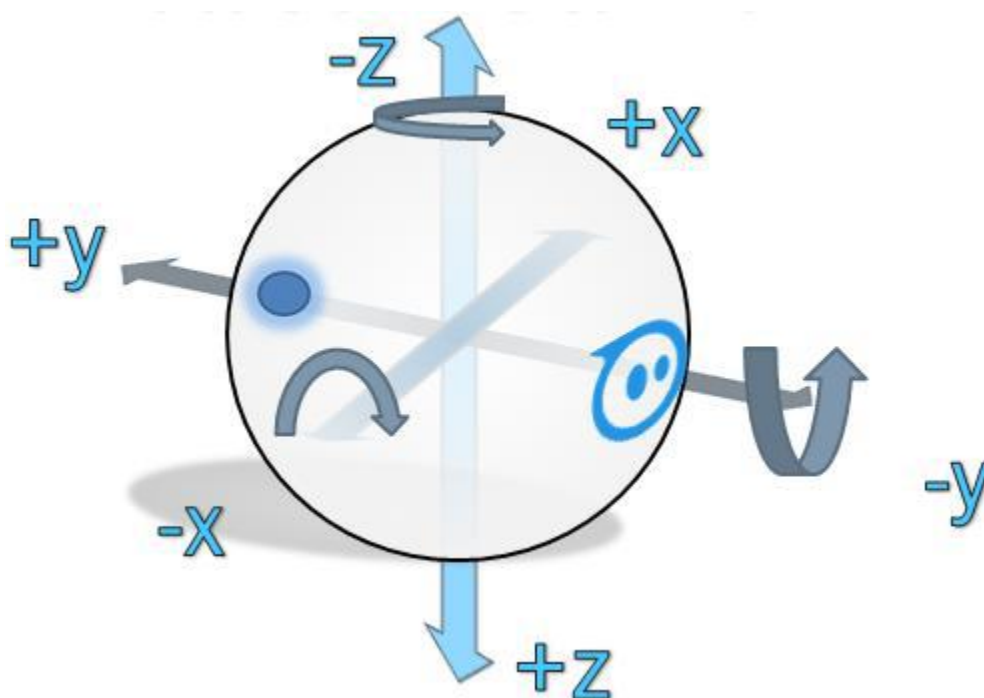


Figura 7.45: Ejes de movimiento del Sphero

Para poder obtener los datos de posición en el LocatorData, se debe crear un Listener asíncrono, de forma que esté continuamente recibiendo la posición del Sphero. De esta forma, cuando se percibe un movimiento, este se dibuja en la pantalla. Las trayectorias deben ser marcadas por la luz de cola. De lo contrario, no se dibujarán correctamente en la pantalla, al menos el objeto deseado:


```
private final DeviceMessenger.AsyncDataListener mDataListener = (data) -> {
    if(data instanceof DeviceSensorsAsyncData){
        List<DeviceSensorsData> data_list = ((DeviceSensorsAsyncData)data).getAsyncData();
        if(data_list != null){
            for(DeviceSensorsData datum : data_list){
                LocatorData locatorData = datum.getLocatorData();
            }
        }
    }
}
```

A la hora de dibujar la trayectoria en la pantalla, se ha elegido la misma solución que en el mando anterior: el paquete android.graphics, en concreto, la clase Canvas. Posee un botón para eliminar la trayectoria y dibujarla de nuevo. Cuando se realiza esta acción, también reiniciarse la posición del Sphero, para que cuando se comience a dibujar la nueva trayectoria, el cursor aparezca en el centro del layout.

```
SetDataStreamingCommand.sendCommand(mRobot, divisor, packet_frames, mask, response_count);
ConfigureLocatorCommand.sendCommand(mRobot, 0, 300, 300, 0);
DeviceMessenger.getInstance().addAsyncDataListener(mRobot, mDataListener);
```

| lastLocation. | | |
|---------------|---------------------------------------|--------------------|
| corx = 1 | getLocatorState () | LocatorSensorState |
| cory = 1 | getPositionY () | float |
| se { | getPosition () | LocatorSensor |
| inicio++ | getPositionX () | float |
| aCorx = | toString () | String |
| aCory = | encode (DeviceMessageEncoder encoder) | void |
| corx = 3 | getVelocity () | LocatorSensor |
| cory = 3 | getVelocityX () | float |
| lastLoca | getVelocityY () | float |
| | equals (Object o) | boolean |

Además de lo comentado anteriormente, también se dispone de un botón para el cambio de mandos.

Para manejar este mando obteniendo resultados positivos hay que tener en cuenta el mecanismo que tiene la pelota en su interior, el cual siempre trata de tener una posición estable. Dada esta situación, si los movimientos a realizar con el Sphero son suaves, el centro de gravedad no cambiará, por lo que se mantendrá estable y solo se detectarán movimientos en

uno de los ejes: X o Y. De esta forma, los datos recibidos en la Tablet PC no son correctos, de forma que se interpreta un movimiento unidimensional.

En el caso de realizar movimientos más fuertes, en los que el mecanismo también gire junto a la carcasa del Sphero, entonces, los datos recibidos son bidimensionales. En algún caso se produce el error del caso anterior, pero son los menos.

La reproducción de los movimientos no está hecha a escala, por lo que su resultado en la pantalla puede ser anecdótico. Sin embargo, tiene justificación debido a que no se puede estar actualizando continuamente el valor de la posición del Sphero, cuando por otro lado, la conexión asíncrona está devolviendo otros datos.

De realizarse actualizando los valores de la posición, modificando manualmente la posición, y viendo una figura más grande en el layout, la serie de errores es muy elevada, por lo que, viendo las posibilidades que ofrece el firmware, y el objetivo de este mando, se ha llegado al máximo de su desarrollo para esta versión del juguete.

Las posibilidades que abre esta opción son infinitas: desde extrapolar la tecnología de posicionamiento de Sphero a otras aplicaciones, hasta aplicar la misma a objetos movibles de forma que limitando sus posiciones no puedan moverse de espacio determinado, pasando por un simple juego de dibujo de trayectorias a utilizar por personas discapacitadas en momentos de aprendizaje.

7.2.2.19 Clase Mando de limitación y corrección de movimientos



Figura 7.46: Pantalla correspondiente al Activity PosicionActivity

El último mando solicitado trataba de resolver los problemas de manejabilidad. Una vez más conviene recordar que esta aplicación está pensada para personas con discapacidad, ya sea motora o intelectual, por lo que se solicitaba la posibilidad de realizar un mando más fácil de manejar que un joystick, y, en la medida de lo posible, implementar medidas de apoyo a sus movimientos, además de limitarlos cuando fuese necesario.

Como solución a esta parte se ha decidido hacer esta clase, que corresponde a la “Activity PosicionActivity”. El aspecto visual es similar a los dos mandos anteriores, es decir, un layout vacío donde el usuario debe dibujar la trayectoria que desea que siga el juguete Sphero. De esta forma ya se consigue facilitar un poco el manejo, al ser más sencillo dibujar que manejar el joystick.

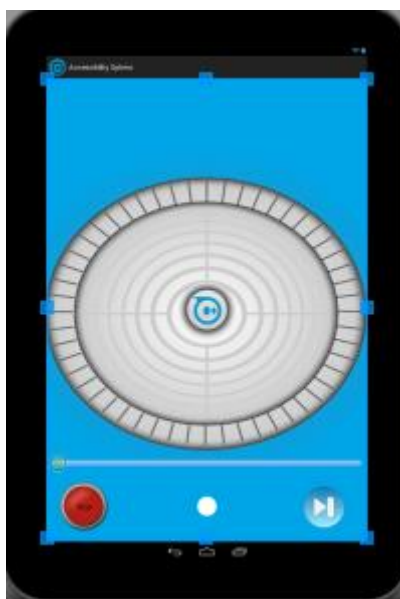


Figura 7.47: Pantalla correspondiente al mando MJoystick

Siguiendo la solicitud de limitar movimientos, se pensó en dividir el layout en cuatro partes, simulando unos ejes de coordenadas y abscisas. De esta forma, existen cuatro cuadrantes, donde el profesor, en su menú, elegirá si el alumno podrá realizar movimientos siguiendo esos ángulos.

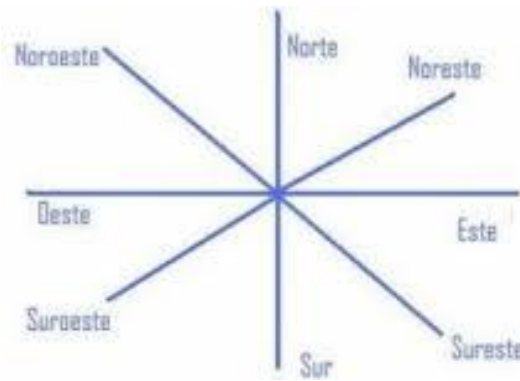


Figura 7.48: Cuatro cuadrantes, con los puntos cardinales que marcan las trayectorias

Por último, y teniendo en cuenta el punto en el que se indica la posibilidad de apoyo a los movimientos, se pensó en realizar un corrector de movimientos, pensando en las posibilidades que puede dar un joystick de una silla de ruedas motorizada. Suponiendo que la persona que está en esa silla puede tener problemas de movilidad, y que en un alto porcentaje de ocasiones la trayectoria que le indica al joystick no es la deseada, se puede poner un punto intermedio, siguiendo los ángulos de ese cuadrante, de forma que hasta ese ángulo se tome una dirección, y el resto de ocasiones, se toma la perpendicular (para cuatro ejes, existe la posibilidad de más ejes, haciendo más efectivo aún el sistema corrector).

```
if(cb090==true && (anguloL >= 0) && (anguloL < 90)){
    if(anguloL< new Float(fCorreccion090)){
        head = 90; // 0, 90 en Sphero
    }else{
        head = 0; //90, 0 en Sphero
    }
}else if(cb90180==true && (anguloL >= 90) && (anguloL < 180)){
    if(anguloL< new Float(fCorreccion90180)){
        head = 270; //90, 270 en Sphero
    }else{
        head = 0; //180, 0 en Sphero
    }
}else if(cb180270==true && (anguloL >= 180) && (anguloL < 270)){
    if(anguloL< new Float(fCorreccion180270)){
        head = 270; //180, 270 en Sphero
    }else{
        head = 180; //270, 180 en Sphero
    }
}else if(cb270360==true && (anguloL >= 270) && (anguloL < 360)){
    if(anguloL< new Float(fCorreccion270360)){
        head = 90; //270, 180 en Sphero
    }else{
        head = 180; // 0, 180 en Sphero
    }
}
```

En la imagen anterior, perteneciente a un trozo de código de esta Activity, se puede observar como la primera comprobación es analizar si está habilitado el cuadrante para que el Sphero se mueva. De no estar activado, el método no devolverá ningún valor de ángulo interpretado por el método de enviar el movimiento al Sphero.

En el caso que esté activado, el segundo paso es analizar si el ángulo de la posición del cursor en la pantalla pertenece a uno u otro cuadrante. Dentro de cada uno de ellos, y siguiendo el ángulo de corrección indicado desde el menú del profesor, obtendrá un ángulo de orientación. Ese es el caso de cuatro cuadrantes, pero puede ampliarse a tantas partes como se desee.

```
if ("cb090S".equals(parameters[0])) {
    cb090 = true;
    fCorreccion090 = parameters[1];
}else if ("cb090N".equals(parameters[0])) {
    cb090 = false;
}

if ("cb90180S".equals(parameters[2])) {
    cb90180 = true;
    fCorreccion90180 = parameters[3];
}else if ("cb90180N".equals(parameters[2])) {
    cb90180 = false;
}

if ("cb180270S".equals(parameters[4])) {
    cb180270 = true;
    fCorreccion180270 = parameters[5];
}else if ("cb180270N".equals(parameters[4])) {
    cb180270 = false;
}

if ("cb270360S".equals(parameters[6])) {
    cb270360 = true;
    fCorreccion270360 = parameters[7];
}else if ("cb270360N".equals(parameters[6])) {
    cb270360 = false;
}
```

A la hora de realizar este mando, existían dos posibilidades: la primera, dibujar una trayectoria, y con el punto final, decidir que ángulo de apertura había, y así, con el factor de corrección, mover el Sphero hacia una trayectoria u otra, o, la segunda, ir enviando movimientos al Sphero en tiempo real, de forma que la trayectoria es corregida en tiempo real. Esta última ha sido a implementada, pensando en las opciones que puede abrir de cara a un futuro.

```
public boolean onTouch(View v, MotionEvent event) {  
  
    corx = (int) event.getX();  
    cory = (int) event.getY();  
  
    dCorx = (double) event.getX();  
    dCory = (double) event.getY();  
  
    heading = getHeadingValue(dCorx,dCory);  
  
    if((heading >= 0) && (heading < 361))  
        RollCommand.sendCommand(mRobot, heading, 0.3f); //VELOCIDAD 0.3  
    else  
        RollCommand.sendStop(mRobot); //Parar en las zonas limitadas  
  
    fondo.invalidate();  
  
    return true;  
}
```

7.2.2.19 Clase Menú ayuda



Figura 7.49: Pantalla perteneciente a la Activity MenuAyuda

La clase corresponde a la “Activity MenuAyuda”, corresponde a una actividad utilizada cuando se pulsa la opción “Ayuda”, del menú de opciones:



Figura 7.50: Acceso al menú de ayuda.

La particularidad que tiene esta clase es que contiene un layout dinámico, de forma que mostrará la información del mando seleccionado en cada momento. Para obtener esa información, existe un flujo de datos entre actividades:

```
intent = new Intent();  
intent.setClass(MJoystick.this, MenuAyuda.class);  
intent.putExtra("nameLayout",  
    getResources().getResourceName(R.layout.ayudajoystick));  
intent.putExtra("idLayout",  
    getResources().getResourceName(R.id.linearAyudaJoystick));  
intent.putExtra("nameController", "Joystick");
```

Así se recupera en el menú Ayuda:

```
Intent intent = getIntent();  
String namelayout = intent.getStringExtra("nameLayout");  
String idlayout= intent.getStringExtra("idLayout");  
nameController= intent.getStringExtra("nameController");
```

De esta forma, el desarrollo de esta actividad es mucho más sencilla que realizar una actividad diferente para cada uno de los mandos. Para esta segunda versión, se han incorporados las modificaciones necesarias para mostrar la ayuda de los cinco nuevos mandos que existen en la aplicación y se ha actualizado la ayuda correspondiente al mando de porciones debido a su cambio de funcionalidad.

Capítulo 8. Anotaciones sobre la primera versión

En este capítulo se van a comentar una serie de anotaciones sobre algún error encontrado en la primera versión y corregido para esta segunda. No era intención localizar estos fallos, pero se ha considerado oportuno solucionarlos para obtener una aplicación con la mayor calidad posible.

8.1 Solución de errores

Durante el proceso de pruebas de la aplicación se ha pretendido localizar cualquier error que esta pudiera contener. Los errores aparecidos de esta nueva versión no son contables como tal, ya que los mismos deben estar corregidos a la hora de la entrega del software. Sin embargo, sí se han localizado tres errores en la primera versión en actividades que no cambiaban su funcionalidad con respecto a esta segunda, y han sido corregidos debido a que producían un cierre abrupto de la aplicación. A continuación, los detalles de los mismos.

8.1.1 Acceso sin datos de usuario

Habitualmente la aplicación tiene unos datos de usuario que han sido introducidos a través del menú del Profesor. Sin embargo, al instalar la aplicación en un dispositivo en el que nunca ha sido instalada previamente, y no disponer del fichero ParamsFlurry.txt, la aplicación comienza a ejecutarse sin un perfilado de usuario. En este caso, si se accedía al menú de alumno sin haber introducido los datos del mismo la aplicación se cerraba.

El error se producía al pasar de la actividad Main a la actividad LaunchController, de forma que al llegar a la nueva actividad, se trataba de cargar los datos a través del fichero nombrado anteriormente y al no encontrarlo se producía un error no controlado. Para evitar que se produjese esta situación, y teniendo en cuenta que los datos de usuario no tienen ningún dato que pueda no cumplir la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, se ha decidido que la actividad Menu controle que sin datos de usuario no se acceda a la siguiente actividad.

```
}else if (!getParamsUserFlurry()){  
    //Modificación para la versión 2: si accedes sin datos del alumno, error y se cierra la aplicación  
    //De esta forma, se controla y se muestra el mensaje de error.  
    Toast.makeText(Main.this,  
        "Indique los datos del alumno en el menú de Profesor",  
        Toast.LENGTH_LONG).show();  
}
```

Figura 8.1: Código para controlar el error por acceso sin datos de usuario

8.1.2 Acceso sin mandos seleccionados

De la misma forma que en el caso anterior, cuando se cambiaba de la actividad Main a la actividad LauncController, si en el menú del Profesor no se habían seleccionado al menos un mando para el manejo del Sphero, la aplicación se cerraba al no estar controlado el error. Para solventar este problema se tomó la misma decisión que en el caso anterior, es decir, controlar en la actividad Main que ha sido seleccionado al menos un mando en el menú del Profesor.

```
case R.id.button2:
    //Modificación para la versión 2: si accedes sin mando seleccionado, error y se cierra la aplicación
    //De esta forma, se controla y se muestra el mensaje de error.
    if(!getList()) {
        Toast.makeText(Main.this,
            "Seleccione al menos un mando en el menú de Profesor",
            Toast.LENGTH_LONG).show();
    }
```

Figura 8.2: Código para controlar el error por acceso sin mandos seleccionados

8.1.3 Botón de añadir mandos

Cuando se accedía al listado de mandos a seleccionar dentro del menú del Profesor, el listado de los mismos aparece oculto, mostrándose solamente los botones Guardar y Añadido/No añadido. En el caso de no mostrar el listado de mandos, si se pulsaba este último botón, la aplicación producía un error con las mismas consecuencias que los dos casos anteriores.

Para solucionarlo se ha controlado que si el controlador del mando es nulo o vacío, el botón continúe mostrando el valor “No añadido” y su estado sea el de “No marcado” (checked = false), de forma que vuelve al estado inicial, sin alterar el funcionamiento de la actividad MenuSelecMandos.

```
case R.id.toggleButtonCheck:
    if(("".equals(addController)) || (addController == null)) {
        toggleButtonCheck.setChecked(false);
        toggleButtonCheck.setTextOff("No Añadido");
    } else {
```

Figura 8.3: Código para controlar el error por pulsar el botón Añadido/No añadido

CONCLUSIONES

Este proyecto surgió como continuación de la primera versión, manteniendo el mismo objetivo, que no es otro que fomentar la interacción con usuarios que presenten una discapacidad. De los objetivos planteados en un inicio, se han cumplido todos ellos:

- Mejorar el control sobre sus movimientos: se ha creado un mando que facilita el uso de la aplicación, al ser un layout sobre el que se dibuja una trayectoria, aplicando los factores de corrección de la misma.
- Limitar movimientos: completando el mando anterior, se pueden limitar los movimientos a realizar.
- Se ha realizado un estudio teórico de elementos externos que pudieran facilitar la interacción con la Tablet PC, y así manejar la aplicación de una forma más sencilla.
- El mando de las porciones de colores ha sido modificado para poder jugar con los mismos y la repetición en el Sphero.
- Otro de los problemas resueltos es el manejo de la aplicación mediante órdenes de voz.
- Además, se ha aprovechado el acelerómetro de la Tablet PC para, mediante los movimientos de la misma, controlar los movimientos del Sphero.
- Se ha eliminado la conexión con Flurry Analytics, reutilizando los datos del perfilado de usuario para el nuevo sistema Adaptare.
- Se ha creado la nueva aplicación web Adaptare, para mostrar las estadísticas de uso de la aplicación. A esta parte le acompaña una nueva base de datos de almacenamiento de los datos de uso, más una serie de webservice para enviarlos desde la aplicación Android a la base de datos.
- Siguiendo el ejemplo del fabricante, se puede acceder a la activación de Bluetooth desde la aplicación en caso de no estar activado.

- Se ha creado un nuevo mando donde se dibuja una trayectoria y es repetida por el juguete Sphero.
- Y en orden inverso, se ha creado otro mando donde el usuario pueda dibujar moviendo la Sphero con la mano a modo de ratón y se muestra en el layout.

De cara a una nueva versión de este proyecto, estas son algunas ideas de mejora que podrían incorporarse a la aplicación:

- Aprovechando el mando por voz, analizar la posibilidad de aceptar sonidos guturales grabados previamente. Muchas personas discapacitadas emiten este tipo de sonidos a la hora de hablar, pero los sistemas de reconocimiento de voz no siempre los tienen en cuenta.
- Además de los sonidos guturales, también se podría diseñar un sistema de ruidos varios, con acciones asociadas, siguiendo combinaciones de los mismos.
- Conectar un dispositivo externo como un joystick vía USB para el manejo de la aplicación, facilitando la interacción con la Tablet PC
- Incorporar una brújula para Android, de forma que Sphero, al recibir las instrucciones de los puntos cardinales, obtenga la dirección correcta marcada por la brújula.
- Realizar una interfaz para, aprovechando algunas de las librerías que hay en el mercado, y la cámara delantera de las Tablet PC, poder realizar al menos la elección de cuatro colores, simulados con botones, con la vista.

- Diseñar y desarrollar un sistema que, aprovechando los mecanismos anticolidión, permita mover al Sphero dentro de unas coordenadas indicadas previamente, deteniéndose en el caso de salir del perímetro indicado previamente.
- En el caso de poder realizarlo con el Sphero 2.0, realizar un estudio práctico de las posibilidades que pueda encontrarse con las nuevas carcasas y rampas.
- Modificación de la velocidad del Sphero, en tiempo real, mediante órdenes por voz, manteniendo el movimiento previamente indicado.

BIBLIOGRAFÍA

➤ **Páginas Web Consultadas**

<http://developer.android.com>

www.xatakandroid.com

www.wikipedia.org

www.niubie.com

<http://www.androidcurso.com/>

www.ausdroid.net

<http://droideando.blogspot.com/>

www.androidpit.es

www.fandroides.com

www.computerhoy.com

www.pcworld.es

www.europapress.es

www.mediamarkt.es

www.matamarcianos.es

<http://www.google.com/glass/start/>

<http://stackoverflow.com/>

<https://github.com>

www.php.net

www.w3schools.com

<http://www.gosphero.com/>

www.flurry.com